Feedback alignment with weight normalization can provide a biologically plausible mechanism for learning

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

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 of forward and backward weights, which is unrealistic given the known connectivity pattern in the brain (known as "weight transport problem"). Notably, it is recently shown that under certain conditions, error BackPropagation Through Random backward Weights (BP-TRW), can lead to partial alignment of forward and backward weights overtime (feedback alignment or FA) and result in surprisingly good accuracies in simple classification tasks using shallow ANNs. In this work, we took a closer look at FA to find out why it occurs when using BP-TRW and explored ways to boost it for deep ANNs. We first show that the gradual alignment of forward and backward weights arises from the successive application of BP-TRW update rule on forward weights regardless of learning or loss function if error signals and outputs of neurons satisfy certain conditions such as when they are autocorrelated. Moreover, we show that FA in deeper networks can be improved significantly by applying a biologically-inspired weight normalization (WN) to the input weights of each neuron. In addition, WN can improve the performance of both BP and BP-TRW when class labels are changed across time, an under-explored phenomenon in ANNs which is crucial for flexible learning in the brain in everyday life. With WN, BP-TRW test accuracy can almost match that of BP following class label changes. Altogether, our results portray a clearer picture of the FA mechanism and provide evidence for how learning can occur using BP-like mechanisms while abiding by biological limits on synaptic weights.

Keywords— feedback alignment, weight transport problem, bio-inspired artificial neural networks, bio-inspired learning methods, biologically-inspired weight normalization, network flexibility

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 not to be plausible in the nervous system [Stork, 1989, Crick, 1989, Song et al., 2020]. One of the key issues in BP which makes it biologically implausible is known as the "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.

Interestingly, 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 possible occurrence of BP-like mechanisms in the brain is suggested [Whittington and Bogacz, 2017, Lillicrap et al., 2020, Xie and Seung, 2003]. In particular, it has been shown that learning occurs even without weight transport by BP-TRW [Lillicrap et al., 2016, Liao et al., 2016], where backward weights are fixed, random and distinct from forward ones. During the learning process of BP-TRW 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. In addition to successive propagation of error in each layer to its previous one in BP-TRW, learning can occur even when error is passed directly from output layer to each hidden layer through random backward weights [Nøkland, 2016, Refinetti et al., 2020]. While there are some investigations on theoretical underpinnings and favorable conditions for FA [Lillicrap et al., 2016, Nøkland, 2016, Refinetti et al., 2020], a thorough examination of mathematical and statistical basis of FA is still lacking.

In this work, we show that FA does not arise from the learning process, optimization, or reduction of loss function; rather, it arises from the successive application of the BP-TRW update rule if error signals and outputs of neurons satisfy certain conditions such as when they are autocorrelated. In general autocorrelation functions (ACFs) of error signals and outputs of neurons play an important role in the final amount of alignment. Furthermore, we show that in deep ANNs,

the accuracy of weight update directions computed by BP-TRW potentially decreases compared to the optimal directions computed by BP as error successively backpropagates towards earlier layers; however, it can be improved by constraining the norm of input weights to each neuron which is a biologically plausible constraint supported by mechanisms like homeostatic synaptic scaling [Turrigiano, 2012], heterosynaptic plasticity [Chistiakova et al., 2015] and intrinsic saturation of each synapse [Bi and Poo, 1998]. We show that supplementing the BP-TRW learning method with WN can improve alignment in deep neural networks and allow for better flexibility in learning new contingencies similar to what is observed behaviorally in biological agents.

2 Results

2.1 Persisting feedback error and input can lead to alignment

For simplicity, consider a two-layer linear ANN with a constant input matrix X where its columns correspond to activity of each input neuron and its rows correspond to each individual input or stimuli. Activity of hidden and output layers are $L_1 = XW_0$ and $L_2 = L_1W_1$, respectively, where W_0 is weight matrix of the first layer, and W_1 is weight matrix of the second layer. With gradient descent, the directions for updating weight matrices computed by BP are $\Delta W_{1,BP} = \eta L_1^T E$ and $\Delta W_{0,BP} = \eta X^T E W_1^T$ where E is the error matrix and η is a constant coefficient (learning rate) [Rumelhart et al., 1985]. With BP-TRW, error backpropagates to the hidden layer by a constant random matrix B, that is, $\Delta W_{0,FA} = \eta X^T E B$ [Lillicrap et al., 2016]. Assuming that the feedback loop is open and a hypothetical constant random E (regardless of loss function and actual error) is fed to backward pass (Fig. 1A) during a period. This assumption is neither a realistic assumption in the learning process of ANNs nor a necessary condition for FA and we will discuss the general case later on, however it gives an initial intuition about the mechanism of FA in ANNs. In this condition, after a large enough number of iterations (k), direction of W_1 converges to

$$W_1[k \gg 1] \simeq c_2 B^T E^T X X^T E \tag{1}$$

where c_2 is a constant coefficient (see Supplementary Note 2). The key factor for alignment of W_1 with B^T is $E^T X X^T E$ as a transformation matrix which applies to B^T (equation 1). Indeed, it is a symmetric semidefinite matrix and this property makes it intrinsically a transformation matrix that tends to partially preserve the direction after transformation (see Supplementary Note 3).

Generally, eigenvalues of a transformation matrix and their arrangement determine the properties of that transformation (Fig. 1B,D). Considere a special case in which elements of $B \in \mathbb{R}^{n_o \times n_h}$, $X \in \mathbb{R}^{n_b \times n_i}$ and $E \in \mathbb{R}^{n_b \times n_o}$ are i.i.d. from $\mathcal{N}(0,1)$ where n_i, n_h, and n_o are the number of neurons in the input, hidden, and output layers, respectively (network dimensions). In this case, by expectation $E^T X X^T E$ resembles an identity matrix scaled by a scaler ($\mathbb{E}(E^T X X^T E) = n_b n_i I$); thus, it is expected to preserve B^T after multiplication, yet the arrangement of eigenvalues, and network dimensions which themselves impact arrangement of eigenvalues, affect the final amount of alignment. For instance, in this case, n_o being more than n_i results in at least $n_o - n_i$ zero eigenvalues that in turn can lead to more deviation and less alignment (Fig. 1B,D, Cond. 2 vs. 1 and 6), increase of n_b improves alignment (Fig. 1B,D, Cond.2 vs. 3) and increase of n_h, which does not contribute in transformation matrix and only appears in the dimensions of B, does not have a significant effect on the mean amount of alignment but decrease the variance of histogram (Fig. 1B,D, Cond. 2 vs. 5, t-test $p > 0.5$).

The final goal of FA is the reduction in $\Delta W_{0,FA} \measuredangle \Delta W_{0,BP}$ and providing a good approximation of gradient direction computed by BP ($\Delta W_{0,BP}$ is only calculated at each iteration for comparison with $\Delta W_{0,FA}$, by which W_0 is actually updated with BP-TRW). In addition to the alignment of W_1 with B^T, alignment between $\Delta W_{0,FA}$ and $\Delta W_{0,BP}$ also happens since in BP we have

$$\Delta W_{0,BP}[k \gg 1] = \eta X^T E W_1^T \simeq \eta c_2 X^T E E^T X X^T E B \tag{2}$$

and in comparison with equation 2 in BP-TRW we have

$$\Delta W_{0,FA}[k \gg 1] \simeq \eta X^T E B. \tag{3}$$

In equations 2 and 3, before substitution of W_1 in equation 2, $X^T E$ can be considered as a transformation matrix applied on two matrices W_1^T and B, which are partially aligned; thus, their transformed are also expected to be partially aligned (later on we will discuss it in deep nonlinear networks). But in here (with the assumption of constant E and X), by substitution of W_1 in equation 2, $X^T E E^T X$ is a positive semidefinite transformation matrix which is applied to $X^T E B$ and can be expected to preserve the direction of $X^T E B$ to some extent after multiplication (Fig. 1C).

2.2 Autocorrrelation of error and input matrix elements leads to alignment

In practical ANNs, updating weights in each iteration changes E continuously. In addition, by using mini-batches, the input matrix changes at each iteration. Even in biological networks, namely visual cortex, input of the network is time-dependent as the projected scene on the retina changes. In this more general case for BP-TRW in a two-layer linear network with actual and variable error $E[k]$ and input $X[k]$ during iterations, expansion of update direction $\Delta W_{1,FA}[k] = \eta L_1[k]^T E[k] = \eta W_0[k]^T X[k]^T E[k]$ by taking successive steps backward along the iterations and substituting W_0, reveals the following terms for $1 \leq o \leq k$ (see Methods)

$$T^o_{1,aln}[k] = \eta^2 B^T E[k-o]^T X[k-o] X[k]^T E[k] \tag{4}$$

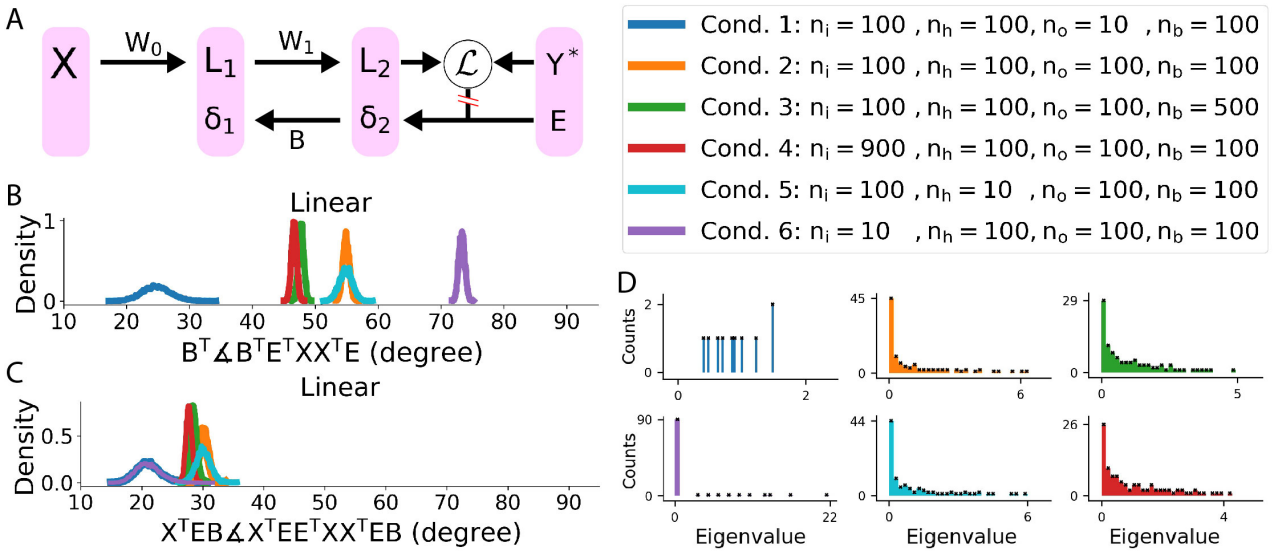

Figure 1 | Alignment occurs in an open-loop network with constant error and input matrices. Based on the formulation of BP-TRW, for a two-layer linear network with constant X and open feedback loop which is fed with constant E, direction of W_1 converges to the direction of $B^T E^T X X^T E$ and if $E^T X X^T E$, as a transformation matrix, partially preserves the direction of B^T after matrix multiplication, alignment occurs, yet the arrangement of its eigenvalues and also network dimensions affect alignment. Accordingly, the amount of alignment is assessed in different conditions where elements of X, E and B are i.i.d. from $\mathcal{N}(0,1)$.
(A) Diagram of an open-loop network in which backward and forward pass is fed with constant E and X respectively.
(B) Histograms of the angle between B^T and $B^T E^T X X^T E$ ($W_1[k \gg 1]$) for a two-layer linear network with constant E and X.
(C) Histograms of the angle between $X^T EB$ ($\Delta W_{0,FA}[k \gg 1]$) and $X^T EE^T XX^T EB$ ($\Delta W_{0,BP}[k \gg 1]$) for a two-layer linear network with constant E and X. This angle may show a different behavior from $B^T \measuredangle B^T E^T X X^T E$ since the transformation matrix applies on $X^T EB$ is $X^T EE^T X$ (Cond.6 panel C vs. B).
(D) Histograms of eigenvalues of $E^T X X^T E$ in different conditions. Eigenvalues of $E^T X X^T E$ as a positive semidefinite transformation matrix play a determinative role in expected final amount of alignment. Network dimensions affect arrangement of eigenvalues and consequently affect the amount that $E^T X X^T E$ preserves the direction of B^T after multiplication. Conditions with more density of eigenvalues near zero result in less alignment between B^T and $W_1[k \gg 1] = B^T E^T X X^T E$.

which we call them alignment term of order o, where $\Delta W_{1,FA}[k] = \eta W_0[0]^T X[k]^T E[k] + T^k_{1,aln}[k] + \cdots + T^2_{1,aln}[k] + T^1_{1,aln}[k]$. Each $T^o_{1,aln}[k]$ propels W_1 towards B^T provided that the transformation matrix $M^o[k] = E[k-o]^T X[k-o] X[k]^T E[k]$ preserves the direction of B^T to some extent after multiplication. In general, $M^o[k]$ is not symmetric but can be decomposed into symmetric and skew-symmetric terms ($M^o[k] = M^o_{sym}[k] + M^o_{skew}[k]$). The skew-symmetric term (or any real skew-symmetric matrix) totally deviates B^T (or any real matrix) after matrix multiplication and $B^T \measuredangle B^T M^o_{skew}[k] = 90°$ (see Methods). Therefore, the two factors which determine the amount of deviation of B^T from $B^T M^o[k]$ are how much M^o_{sym} preserves the direction of B^T after multiplication and the ratio of $\|B^T M^o_{skew}\|_F$ to $\|B^T M^o_{sym}\|_F$.

For alignment of $T^o_{1,aln}[k]$ with B^T, $E[k-o]$, $E[k]$, $X[k-o]$ and $X[k]$ should have some special properties. Each alignment term of order o captures the statistical properties and possible autocorrelation (similarity) of error and input matrix elements in o-step lag and the more $E[k-o]$ and $X[k-o]$ resemble $E[k]$ and $X[k]$, respectively, the more similar $M^o[k]$ is to a symmetric semidefinite matrix. To give an intuition about this, we plotted the histograms of measured angles $B^T \measuredangle T^1_{1,aln}[1]$ (arbitrary $k = 1$, $o = 1$), $B^T \measuredangle B^T M^1_{sym}[1]$, and ratio of $\|B^T M^1_{skew}[1]\|_F$ to $\|B^T M^1_{sym}[1]\|_F$, under various hypothetical conditions (Fig. 2A,B,C). For instance, if the elements of $E[0]$, $E[1]$, $X[0]$, $X[1]$ and B are i.i.d. from $\mathcal{N}(0,1)$ then they are not autocorrelated (in the sense that $\mathbb{E}(X[0]_{i,j} X[1]_{i,j}) = 0$ and $\mathbb{E}(E[0]_{i,j} E[1]_{i,j}) = 0$) and on average no significant amount of alignment is expected (Fig. 2C, Cond. 1, one sample t-test $p > 0.5, n = 10000$). Even if similar to this condition, we generate random $E[0]$, $X[1]$ and $X[0]$ but take $E[1]$ equal to $E[0]$, on average no significant amount of alignment is expected (Fig. 2C, Cond. 2, one sample t-test $p > 0.5, n = 10000$). In another condition, we generated $E[0]$, $E[1]$, $X[0]$ and $X[1]$ with i.i.d. elements from $\mathcal{N}(0.5,1)$ and B with i.i.d. elements from $\mathcal{N}(0,1)$. In this condition, alignment occurs (Fig. 2B,C, Cond. 3) while all elements of X and E are independent yet (positively) autocorrelated during iterations in the sense that

$$\mathbb{E}(X[0]_{i,j} X[1]_{i,j}) > 0, \ \mathbb{E}(E[0]_{i,j} E[1]_{i,j}) > 0.$$

In addition to X and E, the final amount of alignment also depends on the distribution of $B_{i,j}$ (Fig. 2B,C, Cond. 4 vs. 3).

Alignment in the last above condition occurs under the circumstances that $\mathbb{E}(E[1]) = \mathbb{E}(E[0]) = 0.5 J_{n_b \times n_o}$ and $\mathbb{E}(X[1]) = \mathbb{E}(X[0]) = 0.5 J_{n_b \times n_i}$ where $J_{n \times m}$ denotes an $n \times m$ all-ones matrix and by statistical expectation we have $\mathbb{E}(E[0]^T X[0] X[1]^T E[1]) = 0.0625 n_i n_b^2 J_{n_o \times n_o}$. All-ones square matrices are semidefinite with just one positive eigenvalue corresponding to an all-ones eigenvector and the rest of eigenvalues are zero (Fig. 2E, Cond. 6). An all-ones matrix as a transformation matrix also leads to alignment (Fig. 2C, Cond. 6).

The final amount of alignment between B^T and W_1 in learning process is the resultant of all order of alignment terms during iterations (Fig. 2F). Indeed each aligned $T^o_{1,aln}$ injects a component along with B^T into the W_1 and (owing to high-

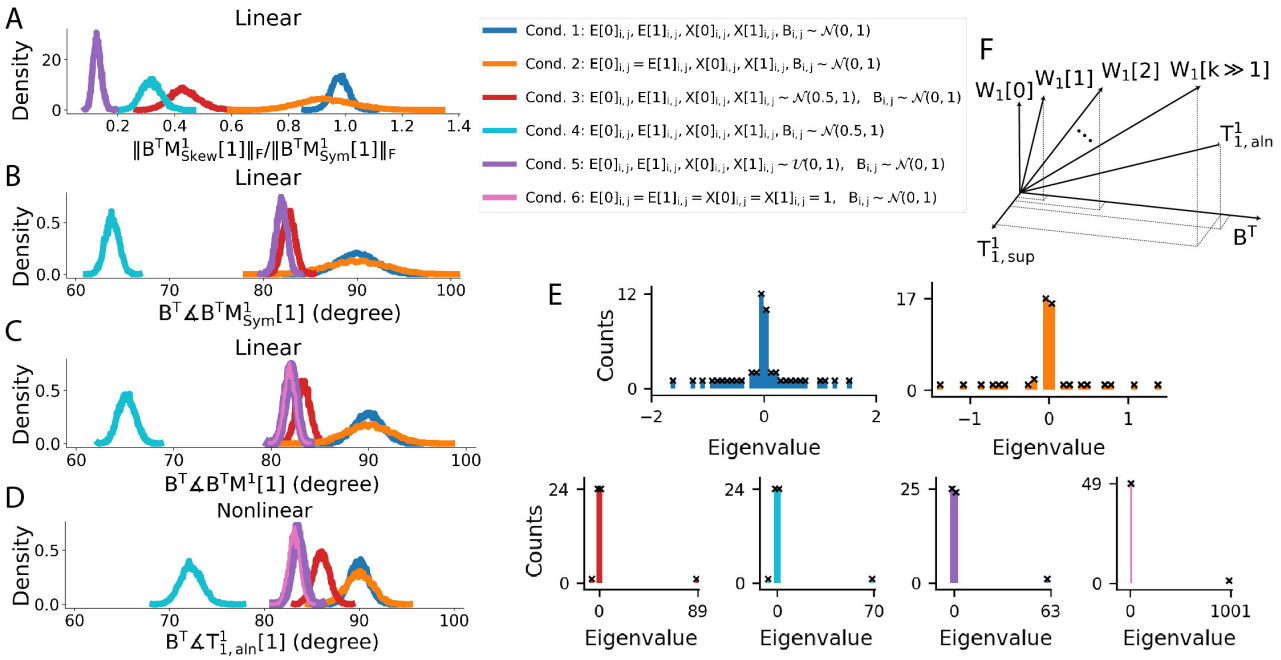

Figure 2 | Alignment terms propel forward weight matrices towards random feedback weights if error and input matrix elements are autocorrelated. To provide an intuition about the FA mechanism in practical ANNs, for a two-layer ANN, the effect of statistical properties of elements of E and X on alignment terms (equation 4) is assessed in different hypothetical conditions. With arbitrary $k = 1$ and $o = 1$, alignment term of interest in linear case (A,B,C) is $T^1_{1,aln}[1] = \eta^2 B^T E[0]^T X[0] X[1]^T E[1]$ and defined transformation matrix is $M^1[1] = E[0]^T X[0] X[1]^T E[1]$. $M^1_{skew}[1]$ and $M^1_{sym}[1]$ are respectively skew-symmetric and symmetric parts of the transformation matrix $M^1[1] = M^1_{sym}[1] + M^1_{skew}[1]$

(A) Histograms of the ratio of $\|B^T M^1_{skew}[1]\|_F$ to $\|B^T M^1_{sym}[1]\|_F$. This ratio is a measure of how much $M^1[1]$ acts like a symmetric transformation on B^T. The more $M^1[1]$ resembles a symmetric matrix, the less this ratio is expected.

(B) Histograms of the angle between B^T and $B^T M^1_{sym}[1]$.

(C) Histograms of the angle between B^T and $B^T M^1[1]$.

(D) Histograms of the angle between B^T and $T^1_{1,aln}[1]$ (equation 5) for a two-layer nonlinear network, in which elements of $f'(\cdot)$ relating to all element-wise matrix multiplications, are 0 or 1 with an equal probability of 0.5. Impact of nonlinearity can be considered as a distortion in linear case formula.

(E) Histograms of eigenvalues of $M^1_{sym}[1]$ in different conditions. If eigenvalues of $M^1_{sym}[1]$ have symmetrical distribution around 0, alignment does not occur (Cond. 1 and 2). $M^1_{sym}[1]$ being positive semidefinite is not a necessary condition for alignment; rather, if its eigenvalues are well arranged, e.g., have positive mean and negative ones are relatively small, alignment is expected (Cond. 3-6).

(F) Diagram of FA mechanism where elements of E are only one step autocorrelated, hence $T^1_{1,aln}$ shows alignment while higher order alignment terms, which are aggregated in the defined auxiliary term $T^1_{1,sup} = T^2_{1,aln} + T^3_{1,aln} + \cdots$ ($\Delta W_{1,FA}[k] = T^1_{1,sup}[k] + T^1_{1,aln}[k]$, see Methods), does not show alignment. The final amount of alignment is resultant of $T^1_{1,aln}$ and $T^1_{1,sup}$. Note that it is a simplified diagram and in practice alignment and sup-alignment terms are variable during iterations. In addition, depending on statistical properties of network matrices, and in particular autocorrelation of elements of X and E in different lags, it may be that contrary to this diagram, $T^1_{1,aln}$ does not show alignment while $T^1_{1,sup}$ shows alignment or both show alignment.

dimensionality and supposing that elements of both X and E can have only non-negative autocorrelation in each lag) each non-aligned $T^o_{1,aln}$ tends to inject W_1 a component which is expected to be perpendicular to B^T (Fig. 2C Cond. 1 and 2) and does not oppose the component which is in line with B^T; however, perpendicular components sum with the aligned ones and reduce their effect in the final amount of alignment.

This analysis can be extended to a nonlinear two-layer network where internal state of hidden and output layers are $Z_1 = XW_0$ and $Z_2 = L_1 W_1$, respectively, and output of layers are $L_1 = f(Z_1)$ and $L_2 = f(Z_2)$ where $f(\cdot)$ is an element-wise activation function and $f'(\cdot)$ is its element-wise derivative and $T^o_{1,aln}[k]$ reads (see Methods)

$$T^o_{1,aln}[k] = \eta\{f'(W_0[k-o]^T X[k]^T) \odot \eta\{f'(Z_1[k-o]) \odot B^T \delta_2[k-o]^T\} X[k-o] X[k]^T\} \delta_2[k] \tag{5}$$

where $\delta_2[k] = E \odot f'(Z_2[k])$ and $\odot$ denotes element-wise matrix multiplication. The difference of a nonlinear network with a linear one is element-wise matrix multiplications and they can be considered as a distortion in the linear $T^o_{1,aln}[k]$ (equation 4). In this regard, assuming B and Z are two independent random matrices with i.i.d. elements from $\mathcal{N}(0,1)$ and proper dimensions, and considering rectified linear unit (ReLU) as activation function, by statistical expectation $\{f'(Z) \odot B\} \measuredangle B$ is $45°$ ($Z, B \in \mathbb{R}^{100\times100}$, $SD = \pm 0.66°$, one sample t-test, $p > 0.5, n = 1000$).

To give an intuition about the impact of these element-wise matrix multiplications on the amount of alignment and compare it with the linear case, we re-plotted the histograms of angles in shallow linear cases (Fig. 2C) by applying nonlinearity (Fig. 2D) with considering ReLU as activation function, which can be roughly considered as the relationship between input current and output firing rate of a biological neuron regardless of saturation, and for each element-wise matrix multiplications of the

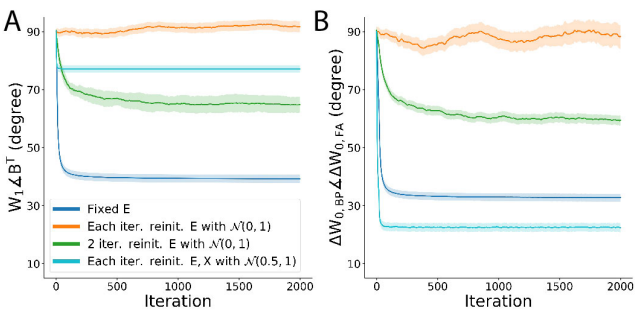

Figure 3 | Autocorrelation of error and input matrix elements leads to alignment. Some hypothetical input and error matrices are imposed on an open-loop nonlinear two-layer ANN.
(A) Angle between forward and backward weights. (B) Angle between BP-TRW weight update direction and gradient direction computed by BP (W_0 is updated by $\Delta W_{0,FA}$, and $\Delta W_{0,BP}$ is only calculated at each iteration for comparison). Legend of (B) is the same as (A).
(blue traces) Elements of E and X are initialized with $\mathcal{N}(0,1)$ and left constant during iterations. According with persisting input and error analysis, alignment occurs.
(orange traces) Elements of E and X are initialized with $\mathcal{N}(0,1)$, and E is reinitialized at each iteration. Alignment does not occur in this case since elements of E are not autocorrelated.
(green traces) Elements of E and X are initialized with $\mathcal{N}(0,1)$, and E is reinitialized at every two iterations. Thus its elements are autocorrelated in half of the iterations, and alignment occurs.
(cyan traces) Elements of E and X are initialized and reinitialized at each iteration with $\mathcal{N}(0.5,1)$. Thus, their elements are autocorrelated and alignment occurs.

equation 5, we independently generated random matrices corresponding to $f'(\cdot)$ in a way that their elements were either 0 or 1 with an equal probability of 0.5. Compared to the linear case, the amount of alignment decreased in this particular nonlinear case but it still happened (Fig. 2C vs. D).

To verify the analysis above and investigate the effect of statistical properties of error and input matrix elements on FA, we imposed some hypothetical input and error matrices (according with update directions of BP-TRW method) on an open-loop nonlinear two-layer ANN in four different cases (Fig. 3). In the first case we initialized elements of E and X with $\mathcal{N}(0,1)$ and kept them constant during iterations. According with persisting E and X analysis, alignment happened (Fig. 3A,B blue traces). In the second case we initialized the network like the first case but reinitialized E after each iteration. In this case, elements of E are not autocorrelated (i.e., $\mathbb{E}(E[k]_{i,j}E[k+o]_{i,j}) = 0, o \neq 0$) and no alignment is expected (Fig. 3A,B orange traces). Compared to the second case, in the third case, we reinitialized E at every two iterations. Thus $E[k-1]$ and $E[k]$ are identical, and consequently, their elements are autocorrelated in half of the iterations and alignment happens (Fig. 3A,B green traces). In the fourth case, we initialized all elements of both E and X with $\mathcal{N}(0.5,1)$ and also reinitialized them at each iteration (with $\mathcal{N}(0.5,1)$). Alignment occurred in this last case (Fig. 3A,B cyan traces) while elements of E and X were independent yet autocorrelated during iterations.

2.3 ACFs of error and input matrix elements and limiting norm of weights affect alignment

In a two-layer ANN, ACFs of both error and input matrix elements play a decisive role in the final amount of alignment and its occurrence since values of them in each nonzero lag o affect the amount of alignment attained and injected to W_1 by $T^o_{1,aln}$. Without putting any constrain, during learning process of ANNs, Frobenius norm of weight matrices can take any value and growth continuously, so-called blow-up, since each aligned $T^o_{1,aln}$ inject a component along with B^T into W_1 in each iteration and accumulation of these aligned components leads to the continuous growth of $\|W_1\|_F$. But mechanisms like homeostatic and heterosynaptic plasticity in the brain prevent such an event from happening. For instance, synaptic scaling has been suggested as a form of homeostatic plasticity which acts at the cellular level and regulates firing rate of neurons by scaling the strength of their input synapses proportionally [Turrigiano, 2012]. Even regardless of synaptic scaling, strong synapses tend to lose their ability of future potentiation, which suggests an upper bound or saturation level for synaptic strengths [Bi and Poo, 1998].

To investigate if accumulation of these aligned components is essential for FA and how limiting norm of weights affects FA, we proposed a strict WN method as an intervention in BP-TRW formula by which we proportionally scale the input weights to each neuron (separately from weights to other neurons) at each iteration so that the Frobenius norm of them became fixed to a non-learnable hyperparameter γ (see Methods). In fact, with this normalization method, we keep all the input weights to each neuron on a hypersphere with radius γ and neglect the weight update direction component that drives out the weights from these neuron-wise hyperspheres. We took γ as a non-learnable hyperparameter since homeostatic processes in the brain act in a non-associative manner compared to other forms of plasticity like Hebbian (see Discussion).

In order to investigate the effect of ACFs of input and error matrix elements and also proposed WN method on FA, we modeled error and input matrix elements with two Gaussian first-order autoregressive stochastic processes and imposed them on forward and backward pass of an open-loop nonlinear two-layer ANN as follows

$$E[k] = E[k-1]\beta + N[k-1](1-\beta) \tag{6}$$

$$X[k] = X[k-1]\beta_X + N_X[k-1](1-\beta_X) \tag{7}$$

where all elements of $N_X[k]$, $X[0]$, $N[k]$ and $E[0]$ are i.i.d. from $\mathcal{N}(0,1)$ and the expansion of ACFs of input and error matrix elements can be controlled by β_X and β, respectively (Fig. 4E).

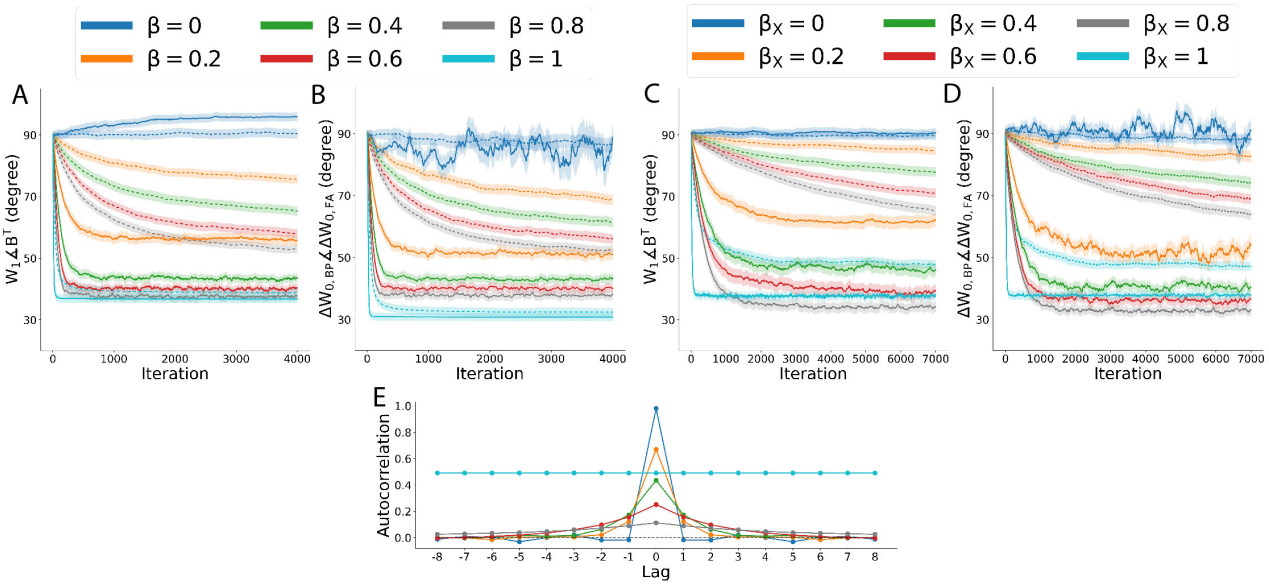

Figure 4 | ACFs of input and error matrix elements and proposed WN method affect alignment. Error and input matrices are modeled with two Gaussian first-order autoregressive stochastic processes (Equ. 6 and 7) in which the expansion of ACF of the input and error matrix elements along lags is controlled by changing β_X and β, respectively, and these input and error matrices are fed to the forward and backward pass of a two-layer nonlinear ANN. In (A,B) $\beta_X = 0$ (X is constant) and in (C,D) $\beta = 0.8$. Solid and dashed traces relate to the cases with and without applying the proposed WN method, respectively.
(A,C) Alignment between forward and transpose of backward weights.
(B,D) Alignment between BP-TRW and BP weight update directions.
(E) Autocorrelograms of single elements of X (or E) for different values of β_X (or β). In extreme cases of $\beta_X = 0$ and $\beta_X = 1$, ACFs are Kronecker delta function and constant function, respectively. For $\beta_X = 0$ or $\beta = 0$, alignment does not occur. Proposed WN method considerably improves FA except for $\beta = \beta_X = 1$, where ACFs of input and error matrix elements are constant function.

In the first case, we kept X constant ($\beta_X = 1$) and changed β from 0 to 1 (Fig. 4A,B dashed-traces). In the second case, we kept $\beta = 0.8$ and changed β_X from 0 to 1 (Fig. 4C,D dashed-traces). In these two cases, β_X and β, had a significant effect on the final amount of alignment.

We re-performed the two previous cases by applying the proposed WN method and fixed the Frobenius norm of input weights of each neuron to $\gamma = 1$ at each iteration. Although it prevents unlimited accumulation of aligned components in W_1, it improved alignment (Fig. 4A,B,C,D solid traces) except where the ACFs of both error and input matrix elements are constant in all lags ($\beta_X = \beta = 1$), and all orders of alignment terms contribute to the alignment equally (Fig. 4A,B cyan traces). For $0 < \beta < 1$ and $0 < \beta_X < 1$, although the ACFs of elements of E and X do not have limited support, their values decay in higher lags; hence higher orders of alignment terms have less contribution to the alignment, which makes WN capable of improving the final amount of alignment (see Supplementary Note 1).

2.4 FA in deep networks and its potential decline in performance

For processing of sensory information in biological neural networks, input signals pass through multiple layers from lower to higher cortical areas. Furthermore, solving complex problems using ANNs also often requires using deep networks. Previously discussed shallow ANN analysis can be generalized to deep nonlinear ANNs where W_ℓ, B_ℓ , $Z_\ell = Z_{\ell-1}W_{\ell-1}$, $L_\ell = f(Z_\ell)$, and $\delta_{\ell,FA}[k] = \delta_{\ell+1,FA}[k]B_\ell \odot f'(Z_\ell[k])$ are respectively forward and backward weight matrix, internal activity, output, and error matrix of neurons in layer ℓ. Similar to shallow ANNs, alignment terms can be extracted from $\Delta W_{\ell,FA}[k] \approx T^1_{\ell,aln}[k] + T^2_{\ell,aln}[k] + \cdots$ (see Methods). Each $T^o_{\ell,aln}$ can propel W_ℓ towards B_ℓ depending on ACFs of elements of $\delta_{\ell+1,FA}$ and $L_{\ell-1}$ in o-step lag, as well as the effect of nonlinearity (element-wise matrix multiplications).

If statistical properties of layers are regulated to be similar, we can expect that $W_\ell \angle B^T_\ell$ be consistent in different layers. But even if this consistency holds for $W_\ell \angle B^T_\ell$ in different layers, $\Delta W_{\ell,FA} \angle \Delta W_{\ell,BP}$ potentially increases as error backpropagates towards earlier layers. For alignment of $\Delta W_{\ell,FA}$ with $\Delta W_{\ell,BP}$ we can write

$$\Delta W_{\ell,FA} = \eta L^T_\ell \delta_{\ell+1,FA} \tag{8}$$

and in comparison with equation 8 for BP we have

$$\Delta W_{\ell,BP} = \eta L^T_\ell \delta_{\ell+1,BP}. \tag{9}$$

The matrix L^T_ℓ is identical in both $\Delta W_{\ell,FA}$ and $\Delta W_{\ell,BP}$, and the factors which determine the angle between them is $\delta_{\ell+1,BP} \angle \delta_{\ell+1,FA}$. Alignment between $\delta_{\ell+1,BP}$ and $\delta_{\ell+1,FA}$ is a limiting factor for the efficiency of BP-TRW learning method compared to BP in deep networks and needs more careful consideration.

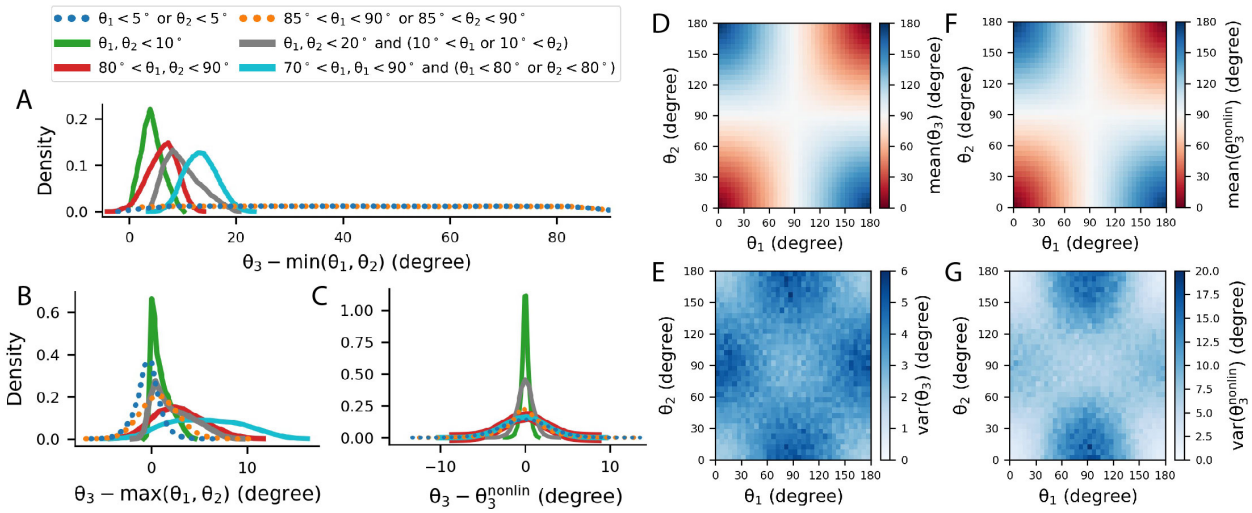

Figure 5 | Successive parallel matrix multiplication of backward and forward weights in deep ANNs, results in a potential increase in deviation of update directions computed by BP-TRW compared to BP.
In a single step of successive parallel matrix multiplication (Equ. 10 vs. 11) where product matrices are B_2B_1 and $W_2^TW_1^T$, B_2 and W_2^T are mutual matrix pairs as well as B_1 and W_1^T, and their mutual cross-angles $\theta_1 = B_1 \measuredangle W_1^T$ and $\theta_2 = B_2 \measuredangle W_2^T$, affects the angle between two product matrices, i.e., $\theta_3 = B_2B_1 \measuredangle W_2^TW_1^T$ in linear case and θ_3^{nonlin} (Equ. 12) in the nonlinear case. For plotting histograms and heatmaps corresponding to θ_3 and θ_3^{nonlin}, the matrices B_1 and B_2 are independently and uniformly drawn from unit hypersphere ($n = 1000000$) and corresponding to each B_1 and B_2, two random matrices W_1^T and W_2^T are drawn from unit hypersphere such that $\theta_1, \theta_2 \sim \mathcal{U}(0°, 180°)$. In the nonlinear case (θ_3^{nonlin}), elements of Z_1 and Z_2 (in Equ. 12) are i.i.d. from $\mathcal{N}(0,1)$ and $f(\cdot) = ReLU(\cdot)$.
(A) Histogram of the difference between θ_3 and minimum of θ_1 and θ_2 in different conditions. Final product matrices are potentially less aligned than the most aligned mutual matrix pairs. If one of the mutual cross-angle of matrices (θ_1 or θ_2) is close to 0° or 90°, $min(\theta_1, \theta_1)$ does not have much contribution in θ_3 (A dotted lines).
(B) Histogram of the difference between θ_3 and maximum of θ_1 and θ_2 in different conditions. In conditions shown with solid lines, final product matrices are potentially less aligned than the least aligned mutual matrix pairs. If one of the mutual cross-angle of matrices (θ_1 or θ_2) is close to 0° or 90°, $max(\theta_1, \theta_1)$ dominantly determines θ_3 (dotted lines).
(C) Histogram of the difference between θ_3^{nonlin} and θ_3. The means of these symmetric histograms are approximately zero; therefore, this applied nonlinearity only increased the variance of angles compared to the linear case.
(D,E) $\theta_1\theta_2$-plane is divided to 5° by 5° bins and heatmap of the average (D) and variance (E) of θ_3 in each bin is plotted.
(F,G) The panels (D) and (E) are re-plotted with applying nonlinearity. This applied nonlinearity does not have a significant effect on the mean of angles compared to linear case (F vs. D) but increases the variance of angles (E vs. G).

To investigate this, for simplicity, consider a d-layer linear ANN where for the last layer ($\ell = d$) we have $\delta_{\ell,FA} = \delta_{\ell,BP}$ but for $0 < \ell < d$ it does not hold; rather, we have

$$\delta_{\ell,FA} = \delta_{d,FA} B_{d-1} B_{d-2} \cdots B_{\ell+1} B_\ell \tag{10}$$

$$\delta_{\ell,BP} = \delta_{d,BP} W_{d-1}^T W_{d-2}^T \cdots W_{\ell+1}^T W_\ell^T. \tag{11}$$

Therefore, owing to this parallel and successive multiplication of forward and backward weights, as error backpropagates towards the initial layers, depending on the mutual cross-angles between corresponding forward and backward weights ($B_\ell \measuredangle W_\ell^T$), deviation of $\delta_{\ell,BP}$ from $\delta_{\ell,FA}$ potentially increase in earlier layers and consequently deviation of $\Delta W_{\ell,FA}$ from $\Delta W_{\ell,BP}$ increase as well and it reduces the efficiency of BP-TRW learning method compared to BP in deep ANNs.

To give an intuition about this, we plotted heat map and histograms of $\theta_3 = B_2B_1 \measuredangle W_2^TW_1^T$ under various conditions (Fig. 5), where we draw two independent random matrices $B_2 \in \mathbb{R}^{30\times15}$ and $B_1 \in \mathbb{R}^{15\times40}$ uniformly from unit hypersphere ($\|B_2\|_F = \|B_1\|_F = 1$) and corresponding to each B_1 and B_2, we draw two independent random matrices $W_2 \in \mathbb{R}^{15\times30}$ and $W_1 \in \mathbb{R}^{40\times15}$ from unit hypersphere in a way 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). In general, there is no explicit inequality between θ_1, θ_2 and θ_3; that is, the amount of θ_3 can be between, more than or less than θ_1 and θ_2 (Fig. 5A,B), but θ_3 being less than both θ_1 and θ_2 is less likely and in here only occurred in the case of $85° < \theta_1, \theta_2 < 90°$ (Fig. 5A, left tail of red histogram). In other words, the amount of alignment between two product matrices B_2B_1 and $W_2^TW_1^T$, tends to be less than the minimum amount of alignment between each W_ℓ^T and B_ℓ. There are also two special cases where if one of the mutual cross angles of matrices, for instance θ_1, is close to 0° or 180°, θ_3 dominantly determines by θ_2 and if one of the mutual cross-angles of matrices, for instance θ_1, is close to 90°, θ_3 tends to be about 90° regardless of θ_2 (Fig. 5A,B,D). In addition to the alignment of $\delta_{\ell,FA}$ with $\delta_{\ell,BP}$, this intuition also justifies alignment of $\Delta W_{\ell,BP}$ with $\Delta W_{\ell,FA}$ (Equ. 8 and 9) from an statistical point of view by considering $\theta_1 = L_\ell^T \measuredangle L_\ell^T = 0$ and $\theta_2 = \delta_{\ell+1,BP} \measuredangle \delta_{\ell+1,FA}$.

This statistical intuition 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 $Z_1 \in \mathbb{R}^{30\times40}$ and $Z_2 \in \mathbb{R}^{30\times15}$ with i.i.d. elements from $\mathcal{N}(0,1)$ and considered $f(\cdot) = ReLU(\cdot)$. Comparing

$$\theta_3^{nonlin} = \{B_2 \odot f'(Z_2)\}B_1 \odot f'(Z_1) \measuredangle \{W_2^T \odot f'(Z_2)\}W_1^T \odot f'(Z_1) \tag{12}$$

with θ_3 (corresponding to the linear case) showed that the main effect of this applied nonlinearity is on the variance of final product angles and it has no significant effect on the means of them (t-test, $p > 0.1$) (Fig. 5C,F,G).

2.5 Proposed WN method can improve alignment in the learning process of a practical deep ANN

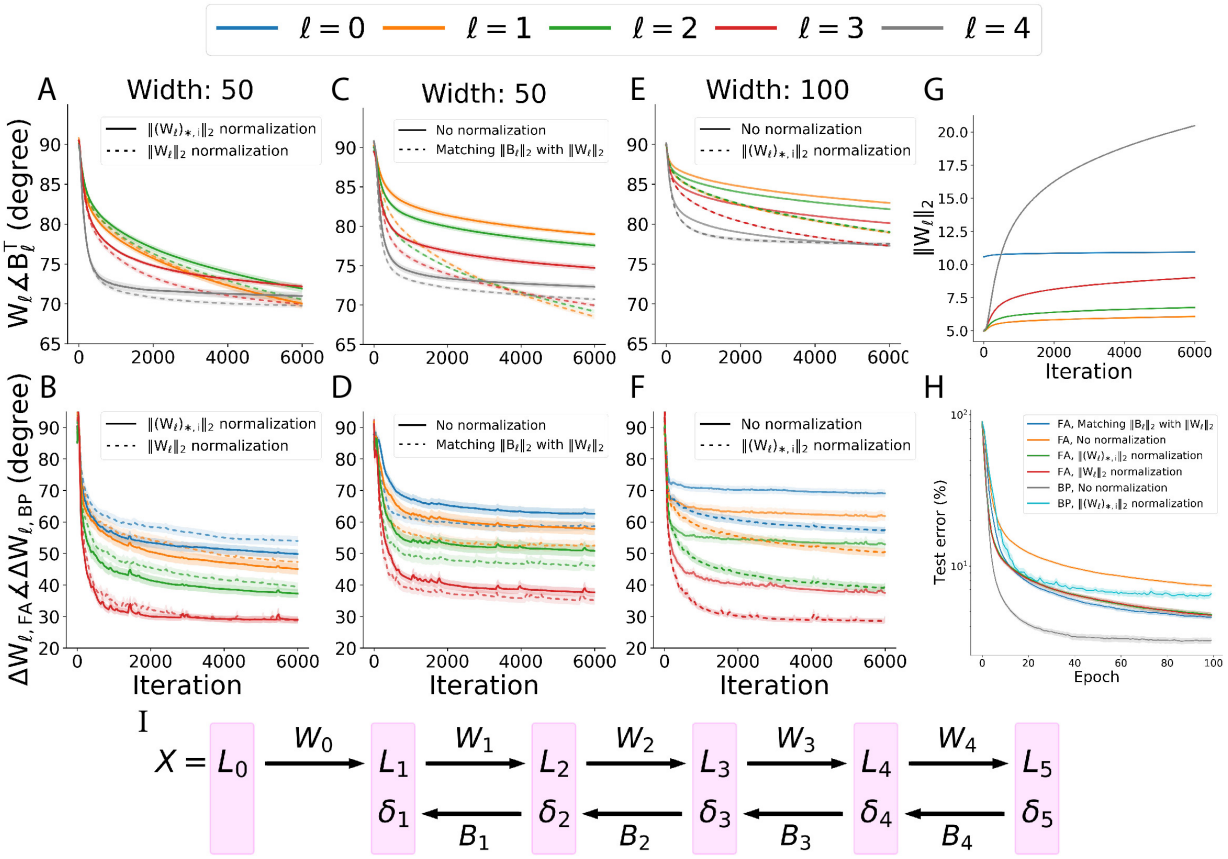

Figure 6 | Increasing network depth or width deteriorates, and proposed WN method improves FA in the learning process of a deep ANN by BP-TRW for handwritten digits classification.
(A,B) A five-layer ANN with 50 neurons in output and hidden layers (network width) is trained by BP-TRW. (solid traces) According to proposed WN method, input weights to each neuron are normalized at every iteration. (dashed traces) The whole weight matrices are normalized at each iteration.
(C,D) A five-layer ANN with 50 neurons in output and hidden layers is trained by BP-TRW. (solid traces) No WN is applied. (dashed traces) $\|B_\ell\|_F$ is matched with $\|W_\ell\|_F$ at every iteration ($B_\ell[k] \leftarrow B_\ell[k] \cdot \|W_\ell[k]\|_F / \|B_\ell[k]\|_F$).
(E,F) A five-layer ANN with 100 neurons in output and hidden layers is trained by FA. (solid traces) No WN is applied. (dashed traces) Input weights to each neuron are normalized at every iteration. Compared to the network with a width of 50, without WN, the amount of alignment decreased and WN overcame the effect of increasing the width to some extent.
(B,D,F) The decrease in alignment between $\Delta W_{\ell,FA}$ and $\Delta W_{\ell,BP}$, as error propagates backward in successive layers, is evident.
(G) Frobenius norm of forward weight matrices corresponds to the learning process of the network with a width of 50 and no WN (C,D solid traces). (H) Test errors of training a network with a width of 50 in different cases. (I) Layout of ANN of this experiment.

In order to study the FA mechanism, the effect of proposed WN method, and potential decline of BP-TRW in practical deep ANNs, as a baseline, we trained a five-layer nonlinear ANN (Fig. 6I) with BP-TRW on MNIST dataset where we divided the training set to 60 mini-batches (Fig. 6C,D solid-traces). To be able to compare alignment in different layers, we matched the number of neurons in hidden and output layers, which we call this number network width, and set it 50 in this baseline ANN. In the last layer, we coded the labels of classes with mutually exclusive 5-hot coding (see Methods). According with previous analysis, as error backpropagates towards earlier layers, successive reduction in accuracy of update directions computed by BP-TRW compared to BP is evident (Fig. 6D solid-traces). The Frobenius norm of forward weight matrices continuously grows, although it is subtle and saturates in earlier layers, it is evident in the last layer (Fig. 6G) and mismatch between the Frobenius norm of forward weights in different layers compared to the Frobenius norm of backward weights, which are constant during the learning process, appears as a distortion in the BP-TRW learning process. In this regard, we matched norm of backward weights with forward weights in each iteration ($B_\ell \leftarrow B_\ell \|W_\ell\|_F / \|B_\ell\|_F$) and with this measure, FA improved (Fig. 6C,D dashed-traces).

Next, we applied proposed weight normalization mechanism to the previous baseline experiment by which Frobenius norm of input weights of each neuron is fixed to $\gamma = 1$ at each iteration (we also applied this WN to B_ℓ at the beginning) and it

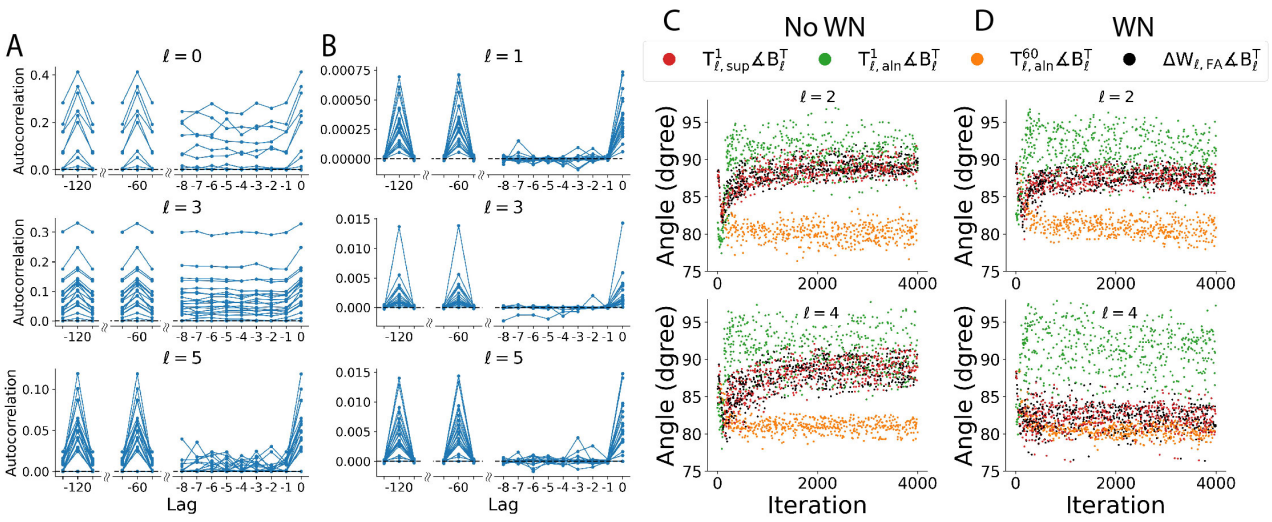

Figure 7 | Taking a closer look at FA in the learning process of a practical ANN on handwritten digits dataset. The behavior of alignment terms depends on the statistical properties and ACFs of error signals ($(\delta_{\ell,FA})_{i,j}$) and outputs of neurons ($(L_\ell)_{i,j}$).
(A,B) Autocorrelograms of some randomly selected elements of L_ℓ (A) and $\delta_{\ell,FA}$ (B) in the learning process of an ANN. In lags that are a multiple of 60, they show more autocorrelation since we used 60 mini-batches.
(C,D) The Behavior of alignment terms of order 1 and 60, sup-alignment term of order 1 ($T^1_{\ell,sup} = T^2_{\ell,aln} + T^3_{\ell,aln} + \cdots$), and ΔW_ℓ with (C) and without (D) applying proposed WN method. At the beginning of the learning process, when neurons are not well trained and do not discriminate between different categories, their error signals have autocorrelation; hence alignment term of order 1 ($T^1_{\ell,aln}[k]$) shows alignment in $k < 250$, but after this initial period, only alignment terms with the order of 60 and its multiples show alignment owing to autocorrelograms of error signals that just show significant autocorrelation in lags that are a multiple of 60.

improved FA (Fig. 6A,B solid-traces). To see if the effect of proposed WN method is only about limiting Frobenius norm of forward weight matrices or not, in another attempt, we normalized the whole weigh matrices ($W_\ell \leftarrow \gamma_\ell W_\ell / \|W_\ell\|_F$ with $\gamma_\ell = 7.07$) at each iteration. Compared to the proposed (neuron-wise) WN method, with this (layer-wise) normalization the alignment between $\Delta W_{\ell,FA}$ and $\Delta W_{\ell,BP}$ slightly decreased (Fig. 6A,B dashed-traces).

To investigate the effect of network width on alignment, we re-performed the baseline experiment with 100 neurons in hidden and output layers. Increasing the width of the network without WN, decreased amount of alignment (Fig. 6E,F solid-traces) and applying proposed WN method with $\gamma = 1$ improved alignment, and to some extend, overcame the effect of increasing the width of the network (Fig. 6E,F dashed-traces).

In addition, we trained the baseline network (with a width of 50) by BP with and without WN to compare final test errors. Although the test accuracy of BP without WN is less than other cases (Fig. 6H dashed-traces), BP-TRW test error is more robust to the proposed WN method (WN decreased test error of BP-TRW, but it increased test error of BP). It has been reported that BP-TRW test error can reach to that of BP in shallow networks for handwritten digits classification task [Lillicrap et al., 2016], but in deep ANNs, potential reduction in the performance of BP-TRW in the sense of $\Delta W_{\ell,BP} \measuredangle \Delta W_{\ell,FA}$, and also test error, is evident and expected (at least without any further consideration) base on our previous analysis.

According to our previous analysis of two-layer ANNs, statistical properties and, in particular, ACFs of outputs and error signals of neurons play a crucial role in FA. Hence, we randomly chose a number of elements of L_ℓ and $\delta_{\ell,FA}$ matrices in the learning process of a five-layer nonlinear ANN on MNIST dataset and plotted autocorrelograms of their activity during iterations (Fig. 7A,B). Since we used 60 mini-batches, elements of $\delta_{\ell,FA}$ only showed considerable autocorrelation in the lags that were a multiple of 60 (Fig. 7B). Therefore, although $T^1_{\ell,aln}$ showed alignment in the initial phase of learning ($k < 250$), when neurons were not trained well and did not discriminate different categories, after that, it did not show significant alignment (Fig. 7C,D green dots) (this holds for all alignment terms which their order is not a multiple of 60). $T^{60}_{\ell,aln}$, which captures autocorrelation of elements of L_ℓ and δ_ℓ in $lag = -60$, showed considerable amount of alignment during all iterations (Fig. 7C,D orange dots), and also $T^1_{\ell,sup} = T^2_{\ell,aln} + T^3_{\ell,aln} + \cdots$, which is the resultant of all alignment terms with order 2 and higher (Fig. 7C,D red dots), showed alignment. Proposed WN method improved amount alignment between $\Delta W_{\ell,FA}$ and B^T_ℓ, and also $T^1_{\ell,sup}$ and B^T_ℓ (Fig. 7C vs. D)

Amount of alignment and test error are relatively robust for a range of γ and we did not make much effort to choose a good γ; however, network is relatively sensitive to it, for example, with small amounts of γ performance of the network is severely reduced (see supplementary Fig. S2 for sensitivity analysis). Even in biological networks, hyperparameters play a crucial role and many diseases, like Alzheimer, are believed to occur owing to the deflection in synaptic strengths [Verret et al., 2012, Frere and Slutsky, 2018, Styr and Slutsky, 2018].

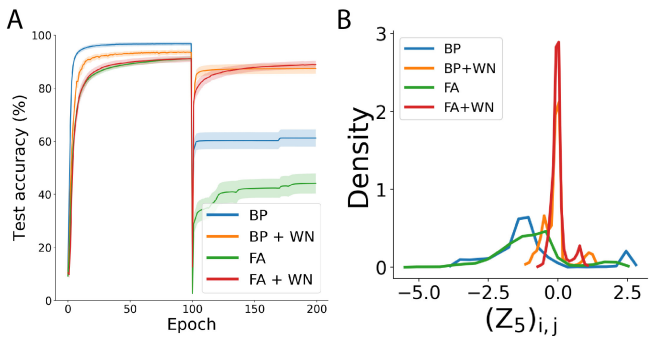

Figure 8 | Proposed WN method improves flexibility of network. Label coding of data is changed after 100 epochs and network should revise the previously learned labels and learn the new ones. (A) Proposed WN method improves the test accuracy after label changing. The network is also trained with both BP-TRW and BP and in this particular case, by applying WN, test accuracy of BP-TRW reached that of BP after changing the label coding. (B) Histogram of the internal state of neurons in the last layer at epoch 100. Proposed WN method regulates distribution of internal states of neurons and it preserves network flexibility for revising. If activity of a neuron saturates, then error signal of that neuron becomes zero ($(\delta_5)_{i,j} = 0$) and input weights to that neuron are no longer updated.

2.6 Proposed weight normalization can improve network flexibility

One of the most prominent features of our brain is its flexibility; that is, it can continuously learn new things or revise wrong concepts that have already been learned. In the field of artificial intelligence, the goal of ANNs is to achieve high test accuracy on a specific dataset in which true labels are known from the beginning of the learning process. But if an ANN is trained with wrong labels or in cases where labels can change in time, there is a question of whether the network can learn the new labels as well as the first ones.

To study this, we trained a five-layer nonlinear ANN on MNIST dataset with BP and BP-TRW for 100 epochs (first phase), then changed the coding of labels (see Methods) and trained it for another 100 epochs (second phase). Without applying the proposed WN method, test accuracy of the second phase did not reach that of the first phase. Therefore, it looks like the network has diminished the ability to revise previously learned labels and learning new ones (Fig. 8A). Interestingly, applying the proposed WN method recovers this ability of the network for both BP and BP-TRW. The effect of WN is evident in the histogram of the internal state of neurons in the last layer at epoch 100 (Fig. 8B). The reason for this is that without WN, activity of most of the neurons saturates ($f'\big((Z_5)_{i,j}\big) = 0$) and error signal of them becomes zero ($(\delta_5)_{i,j} = 0$); thus, input weights to those neurons are no longer updated (so-called gradient vanishing). Note that while without WN, BP-TRW test accuracy after label changing is inferior to BP, applying WN made performance of BP-TRW comparable to that of BP (Fig. 8A). Although we did not too much effort to optimize γ and a range of it can improve network flexibility, yet it is relatively sensitive to network parameters and measures like label smoothing can make it more robust (see supplementary Fig. S3 for sensitivity analysis).

3 Discussion

Statistical basis of FA. Artificial neural networks and their learning paradigms have similarities and differences with biological neural networks. Specifically, backpropagation 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 (reduction of $\Delta W_{\ell,BP}[k] \measuredangle \Delta W_{\ell,FA}[k]$) and provides an approximation of BP.

In this work, we investigated the mathematical and statistical basis of BP-TRW learning method and showed that alignment itself is not due to the learning process or reduction of any loss function; rather, it works as a statistical expectation under certain conditions hence statistical properties of network activities ($f(\cdot)$, Z_ℓ, L_ℓ, δ_ℓ) as well as distribution of weights, affect the occurrence of alignment and its amount. Specifically, in our case studies, ACFs of outputs of neurons ($(L_\ell)_{i,j}$) and their errors ($(\delta_\ell)_{i,j}$) played a crucial role in FA. For instance if the elements of L_ℓ or δ_ℓ were not autocorrelated (their ACF resembled Kronecker delta function), alignment did not occur and if the elements of L_ℓ and δ_ℓ were autocorrelated, alignment happened (Fig. 2, 3, 4). 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 with increasing time lag [Cirillo et al., 2018, Murray et al., 2014, Ogawa and Komatsu, 2010]

A weakness of the BP-TRW learning method as an approximation of BP, which is studied in this work, is the potential decline in accuracy as the depth of the network increases. Indeed, as error backpropagates towards early layers, the angle between $\Delta W_{\ell,BP}$ and $\Delta W_{\ell,FA}$ increases as a consequence of deviation of $\delta_{\ell,BP}$ from $\delta_{\ell,FA}$ (Equ. 11 vs. 10, Fig. 5, Fig. 6B,D,F). However, we note that poor FA in early layers may enhance the capability of the system for unsupervised learning. Indeed, many aspects of the activity of neurons in lower areas of the visual system are demonstrated to be compatible 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].

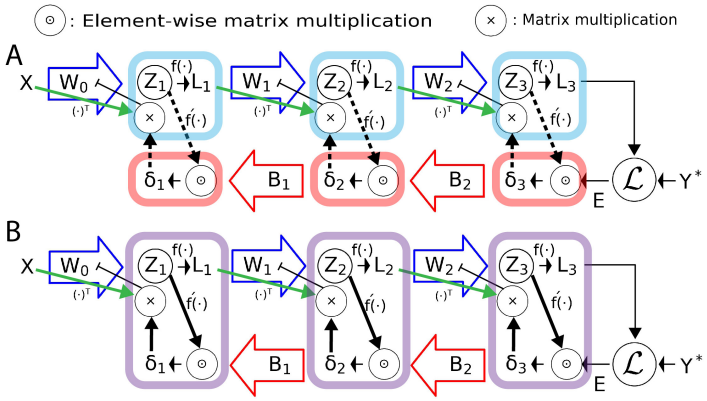

Figure 9 | Visualization of the feedback alignment formula. Two scenarios that can be imagined for occurrence of FA in the brain are illustrated. In these two scenarios, green arrows can be justified by homosynaptic (Hebbian or activity-dependent) plasticity. In BP, B_ℓ is equal to W_ℓ^T which is not biologically plausibles since axons convey information unidirectionally [Shepherd, 2004].
(A) Illustrates the scenario where a distinct parallel feedback network carries the error signals and conveys them to the feedforward network in an interactive way that needs one-to-one reciprocal projections between the neurons of feedforward and feedback network (dashed arrows) and seems biologically implausible.
(B) Illustrates the scenario where distinct parallel feedback weights backpropagate the error signals and convey them through the same feedforward network. This suggests an approximation of operators in purple boxes may be performed in biological neurons with some processes like heterosynaptic plasticity.

Normalization can improve alignment and network flexibility. Normalization is an integral part of the current state of the art ANNs and the correspondence of normalization methods in ANN and biological ones is previously discussed [Shen et al., 2020] and in the context of BP-TRW, the utility of using batch normalization [Ioffe and Szegedy, 2015], is reported [Liao et al., 2016]. There are also numerous reports of normalization mechanisms in biological neural networks working to regulate the activity of neurons and limit synaptic weights dynamic ranges [Turrigiano, 2012, Chistiakova et al., 2015, Bi and Poo, 1998].

Accordingly we proposed a strict WN method, which is done by constraining the Frobenius norm of input weights of each neuron to a limited amount at each iteration, and it improved FA (Fig. 6, Fig. 4) and also flexibility of network (Fig. 8) by enhancing the network capability of revising previously learned labels. We showed that this improvement is due to keeping the neuronal activations away from the extremes of activity range (Fig. 8B) and avoiding the gradient vanishing problem that can otherwise hamper new learning. We also considered the amount to which norm of weights is fixed (γ) a non-learnable hyperparameter of the network as homeostatic plasticity in the brain acts in a non-associative 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 non-associative and competitive manner, i.e., potentiation of a synapse can result in depression of another synapse [Chistiakova et al., 2015].

The impact of nonlinearity on the FA depends on the activity of neurons as well as the distribution of weights. Not only in BP-TRW, but also in BP $f'(Z_\ell)$ is important. For instance if most of the neurons are inactive or saturated (in the case of using activation functions like $tanh(\cdot)$), then for them we have $f'((Z_\ell)_{i,j}) \simeq 0$ and the error signals do not backpropagate well (so-called gradient vanishing) and if most of them are regulated such that $f'((Z_\ell)_{i,j}) \simeq c$, where c is a constant value, then nonlinearity impose little distortion compared to the linear case. Therefore in both BP-TRW and BP, regulation of activity of neurons is important. Even in the biological neurons, homeostatic mechanisms which regulate activity of neurons and prevent them from high or low firing rates have been reported [Murthy et al., 2001, Surmeier and Foehring, 2004, Turrigiano, 2012].

Approximation of BP in biological networks Weight transport problem is only one of the biological implausibilities of BP formula which can be avoided by using BP-TRW methods. There are a bunch of other biological implausibilities in both BP and BP-TRW [Marblestone et al., 2016, Stork, 1989]. For instance, firing rate as 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 output of artificial neurons. Indeed they are internal attributes of neurons that backpropagate to other neurons through some putative feedback synapses, 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 while 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]. Furthermore, there is a debate about whether the computational atoms in the neural coding are spike timings or firing rates [Brette, 2015]. Obviously, the current formulation of BP-TRW (or BP) is only compatible with the rate-based models rather than the spike-based models.

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, Whittington and Bogacz, 2017, Lillicrap et al., 2020]. According to this, we visualized the BP-TRW formula and imagined two different scenarios for the occurrence of FA in the brain (Fig. 9). In the first one, the error signals backpropagate through some other neurons (or a parallel network) distinct from feedforward ones (Fig. 9A) similar to what has been proposed previously [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 path (Fig. 9A dashed arrows) which does not seem biologically plausible. In the second scenario, the error signals backpropagate to the same neurons of the forward path through some other axons and synapses distinct from feedforward ones (Fig. 9B). If we assume that an approximation of BP-TRW occurs in the brain, this latter proposed scenario suggests

that an approximation of the calculation of internal $\delta_{\ell+1}$ may occur locally [Guerguiev et al., 2017] (Fig. 9B purple boxes), and the operation of $\Delta W_\ell = \eta L_\ell^T \delta_{\ell+1}$ can be justified by homosynaptic (Hebbian or activity-dependent) plasticity.

In the BP-TRW learning method, the B_ℓ 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 approximate $\delta_{\ell+1}$ as an internal attribute of biological neurons, the operation of $\Delta W_\ell = \eta L_\ell^T \delta_{\ell+1}$ is performed approximately as heterosynaptic plasticity between forward and backward synapses afferent to 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 ($\sum_{i,j}(W_\ell)_{i,j}(B_\ell)_{j,i} > 0$, supposing two consecutive artificial neural layers, $(W_\ell)_{i,j}$ is a feedforward connection from neuron i in the first layer to neuron j in the second layers and $(B_\ell)_{j,i}$ is its reciprocal connection from neuron j in the second layer to neuron i in the first layer). 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 non-random reciprocal connectivity, is reported among pyramidal cells [Song et al., 2005, Markram et al., 1997, Holmgren et al., 2003, Sjöström et al., 2001]. Although these non-random 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 and the provided intuitions portray a clearer picture of the FA mechanism when BP-TRW learning method is used 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 equipped with WN was capable of approximating the optimal weight changes proposed by BP in simple feedforward networks, 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 comparative performance of BP vs. BP-TRW. In addition to the fully connected architectures considered here, BP-TRW in convolutional or recurrent networks that are both biologically relevant remains to be examined in the future.

4 Methods

4.1 BP and BP-TRW learning methods

For conventional d layer ANNs, we denoted weight matrices, internal state of neurons, and output 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. For $0 < \ell \leq d$, we calculated internal state of neurons in layer ℓ according with $Z_\ell = L_{\ell-1}W_{\ell-1}$. In a more general case a bias vector $\mathbf{b}_\ell \in \mathbb{R}^{1 \times n_\ell}$ can be added to Z_ℓ ($Z_\ell = L_{\ell-1}W_{\ell-1} + \mathbf{b}_\ell$) where addition of a matrix with a row vector is defined as adding the vector to each row of the matrix, but for simplicity, we did not consider $\mathbf{b}_\ell$ in our experiments and also analysis since it can be embedded in $W_{\ell-1}$ by adding a corresponding all-ones column to the $L_{\ell-1}$. We denoted input, output, and desired output matrix (true labels) of ANNs by $X = L_0 \in \mathbb{R}^{n_b \times n_i}$, $Y = L_d \in \mathbb{R}^{n_b \times n_o}$, and $Y^* \in \mathbb{R}^{n_b \times n_o}$, respectively, where $n_i = n_0$ is number of neurons in the input layer, and $n_o = n_d$ is number of neurons in the output layer. For two-layer ANNs we denoted number of neurons in the only hidden layer by $n_h = n_1$.

In the experiments where we trained an ANN on a specific task (MNIST dataset for handwritten digits classification, Fig. 6, Fig. 7, Fig. 8), we calculated error matrix at each iteration k according with $E[k] = Y^*[k] - Y[k]$ and the loss function was $\mathcal{L} = \frac{1}{2}\sum_i \sum_j E_{i,j}^2$ where $E_{i,j}$ is the element in i^{th} row and j^{th} column of matrix E.

In BP learning method, we updated weights at each iteration as below

$$W_\ell[k+1] = W_\ell[k] + \Delta W_\ell[k] \tag{13}$$

where the direction of the gradient for updating weight matrices computed by BP at each iteration k is

$$\Delta W_{\ell,BP}[k] = -\eta \left.\frac{\partial \mathcal{L}}{\partial W_\ell}\right|_k = \eta L_\ell[k]^T \delta_{\ell+1,BP}[k],\ 0 \leq \ell < d \tag{14}$$

where

$$\delta_{d,BP}[k] = E[k] \odot f'(Z_d[k]) \tag{15}$$

$$\delta_{\ell,BP}[k] = \delta_{\ell+1,BP}[k] W_\ell[k]^T \odot f'(Z_\ell[k]),\ 0 < \ell < d \tag{16}$$

and $\odot$ denotes element-wise matrix multiplication (in the order of operations, it has less priority than matrix multiplication), $f'(\cdot)$ 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 backpropagates by constant random matrices different from forward weights which we denoted by $B_\ell \in \mathbb{R}^{n_{\ell+1}\times n_\ell}$ (in two-layer ANNs we denoted $B = B_1$), and calculated update directions at each iteration as follows (W_ℓ^T in equation 16 is replaced with B_ℓ)

$$\delta_{d,FA}[k] = \delta_{d,BP}[k] = E[k] \odot f'\big(Z_d[k]\big) \tag{17}$$

$$\delta_{\ell,FA}[k] = \delta_{\ell+1,FA}[k]B_\ell \odot f'\big(Z_\ell[k]\big),\ 0 < \ell < d \tag{18}$$

$$\Delta W_{\ell,FA}[k] = \eta L_\ell[k]^T \delta_{\ell+1,FA}[k],\ 0 \le \ell < d. \tag{19}$$

In the experiments where we intended to investigate statistical basis of FA, we imposed hypothetical random X and E on an open-loop network regardless of loss function.

The final goal of BP-TRW is providing a good approximation of $\Delta W_{\ell,BP}[k]$, i.e., reduction in $\Delta W_{\ell,FA}[k] \measuredangle \Delta W_{\ell,BP}[k]$. According to this, 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 for updating forward weight matrices.

4.2 Derivation of alignment terms and their corresponding transformation matrices

To see why BP-TRW works and providing an intuition about its statistical basis without imposing any unrealistic assumption (e.g., $XX^T = I$, zero initialized weights, or constant error and input matrices), we expanded $\Delta W_{\ell,FA}[k]$ by taking successive steps backward along the iterations and substituting every $W_0[k-o]$ for $0 \le o < k$, and by applying Taylor approximation as below

$$\begin{aligned}
\Delta W_{\ell,FA}[k] &= \eta L_\ell[k]^T \delta_{\ell+1,FA}[k] = \eta f\big(W_{\ell-1}[k]^T L_{\ell-1}[k]^T\big)\delta_{\ell+1,FA}[k] \\
&= \eta f\big(\{W_{\ell-1}[k-1]^T + \eta\delta_{\ell,FA}[k-1]^T L_{\ell-1}[k-1]\}L_{\ell-1}[k]^T\big)\delta_{\ell+1,FA}[k] \\
&\approx \eta\big\{f\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T\big)\ + \\
&\quad f'\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T\big) \odot \eta\delta_{\ell,FA}[k-1]^T L_{\ell-1}[k-1]L_{\ell-1}[k]^T\big\}\delta_{\ell+1,FA}[k] \\
&= T^1_{\ell,aln}[k] + T^1_{\ell,sub}[k] \\
&\approx T^1_{\ell,aln}[k] + T^2_{\ell,aln}[k] + \cdots + f\big(W_{\ell-1}[0]^T L_{\ell-1}[k]^T\big)\delta_{\ell+1,FA}[k],\ 0 < \ell < d
\end{aligned} \tag{20}$$

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

$$\begin{aligned}
&T^o_{\ell,aln}[k] = \eta\big\{f'\big(W_{\ell-1}[k-o]^T L_{\ell-1}[k]^T\big) \odot \eta\delta_{\ell,FA}[k-o]^T L_{\ell-1}[k-o]L_{\ell-1}[k]^T\big\}\delta_{\ell+1,FA}[k] = \\
&\eta\big\{f'\big(W_{\ell-1}[k-o]^T L^T_{\ell-1}[k]\big) \odot \eta\{f'\big(Z_\ell[k-o]\big)^T \odot B_\ell^T \delta^T_{\ell+1,FA}[k-o]\}L_{\ell-1}[k-o]L_{\ell-1}[k]^T\big\}\delta_{\ell+1,FA}[k]
\end{aligned} \tag{21}$$

as alignment term of order o in layer ℓ, and for $1 \le o < k$ and $0 < \ell < d$ we define

$$\begin{aligned}
T^o_{\ell,sup}[k] &= \eta f\big(W_{\ell-1}[k-o]^T L_{\ell-1}[k]^T\big)\delta_{\ell+1,FA}[k] \\
&\approx T^{o+1}_{\ell,aln}[k] + T^{o+2}_{\ell,aln}[k] + \cdots + f\big(W_{\ell-1}[0]^T L_{\ell-1}[k]^T\big)\delta_{\ell+1,FA}[k]
\end{aligned} \tag{22}$$

as sup-alignment term of order o in layer ℓ which itself contains alignment terms of order 2 and higher (we defined it for the purposes of summarizing and illustration).

Depending on the number of layers, and whether or not nonlinearity is applied, alignment terms can be summarized and represented. For a linear network, where $f(\cdot)$ is an all-ones matrix, element-wise matrix multiplications are eliminated and (without Taylor approximation) we have

$$\Delta W_{\ell,FA}[k] = T^1_{\ell,aln}[k] + T^2_{\ell,aln}[k] + \cdots + W_{\ell-1}[0]^T L_{\ell-1}[k]^T \delta_{\ell+1,FA}[k] \tag{23}$$

where alignment term of order o in layer ℓ is

$$T^o_{\ell,aln}[k] = \eta^2 B_\ell^T \delta_{\ell+1,FA}[k-o]^T L_{\ell-1}[k-o]L_{\ell-1}[k]^T \delta_{\ell+1,FA}[k] \tag{24}$$

In two-layer (shallow) nonlinear ANNs with $B = B_1$, alignment term of order o is

$$\begin{aligned}
&T^o_{1,aln}[k] = \eta\big\{f'\big(W_0[k-1]^T X[k]^T\big) \odot \eta\delta_1[k-1]^T X[k-1]X[k]^T\big\}\delta_2[k] = \\
&\eta\big\{f'\big(W_0[k-o]^T X[k]^T\big) \odot \eta\{\delta_2[k-o]B \odot f'\big(Z_1[k-o]\big)\}^T X[k-1]X[k]^T\big\}\delta_2[k]
\end{aligned} \tag{25}$$

which can be summarized in two-layer linear ANNs as below

$$T^o_{1,aln}[k] = \eta^2 B^T E[k-o]^T X[k-o]X[k]^T E[k]. \tag{26}$$

Accordingly, the transformation matrix which applies to B^T in two-layer linear ANNs is

$$M^o[k] = E[k-o]^T X[k-o]X[k]^T E[k] \tag{27}$$

In general $M^o[k]$ is not symmetric but we can split it into two symmetric and skew-symmetric terms

$$M^o[k] = M^o_{sym}[k] + M^o_{skew}[k] \tag{28}$$

where

$$M^o_{sym}[k] = \frac{1}{2}(M^o[k] + M^o[k]^T). \tag{29}$$

and

$$M^o_{skew}[k] = \frac{1}{2}(M^o[k] - M^o[k]^T) \tag{30}$$

The skew-symmetric term (or any real skew-symmetric matrix), totally deviates B^T (or any real matrix) after matrix multiplication since

$$\begin{aligned} \langle B^T, B^T M^o_{skew}[k] \rangle_F &= tr(B^T M^o_{skew}[k] B) = tr(B^T M^o_{skew}[k]^T B) \\ &= -tr(B^T M^o_{skew}[k] B) = 0 \end{aligned} \tag{31}$$

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

4.3 Network parameters, dimensions, and initialization

For plotting histograms of Fig. 2 we chose network dimensions $n_i = n_h = 100$, $n_b = 20$, and $n_o = 50$.

In experiments on two-layer ANNs (Fig. 3 and Fig. 4) we chose network dimensions $n_i = n_h = n_b = 100$ and $n_o = 10$, and set learning rate to $\eta = 0.003$. We chose n_o relatively small for illustrative reasons since increasing it without increase n_i, decreases the amount of alignment (Fig. 1). In these experiments, we initialized elements of B, W_0 and W_1 with i.i.d. random variables from $\mathcal{N}(0, 1)$ and used $ReLU(\cdot)$ as activation function ($f(\cdot) = ReLU(\cdot)$ and for its element-wise derivative $f'(\cdot)$ we considered $f'(0) = 1$ by convention). For proposed WN method we set $\gamma = 1$ in these experiments.

For training deep ANNs on handwritten digits dataset (MNIST), in the experiments of Fig. 6 and Fig. 7 we set learning rate to $\eta = 0.005$, and in the experiment of Fig. 8 we set learning rate to $\eta = 0.003$. In these experiments we initialized elements of B, W_0 and W_1 with i.i.d. random variables from $\mathcal{N}(0, 0.1)$ and for nonlinearity we used $f(\cdot) = tanh(ReLU(\cdot))$ as activation function. Number of input neurons in these experiments was 225 since we resized all handwritten digits to 15×15 and then vectorized them. We also normalized intensity of their pixels (output of each input neuron) to lay between 0 and 1 (dividing them by 255). For proposed WN method we set $\gamma = 1$ in the experiments of Fig. 6 and Fig. 7, and $\gamma = 0.6$ in the experiments of Fig. 8. In the experiments of Fig. 7 and Fig. 8 width of the network (number of neurons in each hidden and output layer) was 50.

4.4 Proposed weight normalization mechanism

We kept the Frobenius norm of input weights to each neuron constant by the following operation

$$W_\ell[k]_{*,i} \leftarrow \gamma \frac{W_\ell[k]_{*,i}}{\|(W_\ell[k])_{*,i}\|_F} \tag{32}$$

which performs at each iteration of learning process for every column of all weight matrices ($W_\ell[k]_{*,i}$ denotes a matrix consist of i^{th} column of $W_\ell[k]$ which corresponds to the all input weights to the neuron i in the layer $\ell + 1$). Whenever we used proposed WN method, we also normalized every columns of B_ℓ with γ at the beginning of learning process(after network initialization).

4.5 Angle between two matrices.

We calculated the angle between two arbitrary matrices W and B with the same dimensions as follows:

$$W \measuredangle B = cos^{-1}\left(\frac{\langle W, B \rangle_F}{\|W\|_F \|B\|_F}\right) \tag{33}$$

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 vector angle between vectorized W and B in the euclidean space. The angle between two matrices is indeed a measure for the similarity between the normalized version of two matrices (regardless of $\|W\|_F$ and $\|B\|_F$).

4.6 Calculating ACF for correlograms.

In figures 4E and 7A,B we calculated the ACF of a signal $S[k]$ with length N ($1 \leq k \leq N$) in each lag $-N < \tau < N$ by

$$ACF(\tau) = \frac{1}{N - |\tau|} \sum_{k'=1}^{N-|\tau|} S[k'] S[k' + |\tau|]. \tag{34}$$

4.7 Plotting the results.

In all figures where traces corresponding to $\Delta W_{\ell,FA} \measuredangle \Delta W_{\ell,BP}$ is illustrated, owing to the noise of the original traces, we first passed traces from a moving average filter with a length of 60 and then plotted them. In all figures, shaded areas are $\pm SEM$ (standard error of the mean). In figures 3 and 4 each trace is average of 30 repetitions. In figures 6 and 8 each trace is average of 10 repetitions.

4.8 Generating two random matrices from unit hypersphere with uniform angle.

In figure 5, for generating two random matrices from the unit sphere with uniform angle ($B_\ell \measuredangle W_\ell^T \sim \mathcal{U}(0°, 180°)$), 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 ($W_\ell^T \leftarrow W_\ell^T / \|W_\ell^T\|_F$).

4.9 Generating mutually exclusive n-hot coding.

In training deep ANNs on handwritten digits dataset (MNIST) (Fig. 6 and Fig. 8) where network width was 50 we used mutually exclusive 5-hot coding and where network width was 100 we used mutually exclusive 10-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. For network flexibility analysis (Fig. 8) we repeated this procedure at the epoch 100 and generated new coding vectors.

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

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

[Bi and Poo, 1998] 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.

[Brette, 2015] Brette, R. (2015). Philosophy of the spike: rate-based vs. spike-based theories of the brain. *Frontiers in systems neuroscience*, 9:151.

[Cadieu et al., 2014] 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 et al., 2015] 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 et al., 2016] 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 et al., 2018] 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 et al., 2010] 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.

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

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

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

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

[Holmgren et al., 2003] 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.

[Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

[Khaligh-Razavi and Kriegeskorte, 2014] 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 et al., 2018] 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.

[Liao et al., 2016] 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 et al., 2016] 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 et al., 2020] 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 et al., 2016] 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 et al., 1997] 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.

[Murray et al., 2014] 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 et al., 2001] 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 et al., 2018] 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. *Advances in Neural Information Processing Systems*, 31:5290–5301.

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

[Ogawa and Komatsu, 2010] 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 and Field, 1996] 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 et al., 2020] Refinetti, M., d'Ascoli, S., Ohana, R., and Goldt, S. (2020). The dynamics of learning with feedback alignment. *arXiv preprint arXiv:2011.12428*.

[Rumelhart et al., 1985] 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.

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

[Shepherd, 2004] Shepherd, G. M. (2004). *The synaptic organization of the brain*. Oxford university press.

[Sjöström et al., 2001] 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 et al., 2005] 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 et al., 2020] 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, 1989] Stork, D. G. (1989). Is backpropagation biologically plausible. In *International Joint Conference on Neural Networks*, volume 2, pages 241–246. IEEE Washington, DC.

[Styr and Slutsky, 2018] 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 and Foehring, 2004] Surmeier, D. J. and Foehring, R. (2004). A mechanism for homeostatic plasticity. *Nature neuroscience*, 7(7):691–692.

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

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

[Verret et al., 2012] 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 and Bogacz, 2017] 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 and Bogacz, 2019] Whittington, J. C. and Bogacz, R. (2019). Theories of error back-propagation in the brain. *Trends in cognitive sciences*, 23(3):235–250.

[Xie and Seung, 2003] 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 and Andersen, 1988] Zipser, D. and Andersen, R. A. (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, 331(6158):679–684.

Supplementary Information

Supplimentory Note 1. Behavior of alignment and sup-alignment terms in FA with applying proposed WN method in Shallow Nonlinear ANNs

Based on our analysis, if ACFs of error matrix elements resemble a Kronecker delta function, alignment does not occur. By expansion of its support to further lags, more orders of alignment terms contribute in alignment. Assume that in a two-layer ANN, input matrix is constant and ACFs of error matrices elements have limited support of $lag \in [-s, s]$ and show significant (positive) autocorrelation for $-s \leq lag \leq s$. With this assumption, T^o_{aln} with $o \leq s$ contribute in alignment and

$$T^s_{\ell,sup}[k] = \eta f\big(W_{\ell-1}[k-o]^T L_{\ell-1}[k]^T\big)\delta_{\ell+1,FA}[k] \tag{S1}$$

does not contribute in alignment. In a two-layer network, with an increasing activation function, $\|W_0\|_F$ directly affects $\|T^s_{1,sup}\|_F$ while if we neglect its indirect effect on $\|E\|_F$ (assuming that the neurons maintain the magnitude of their error signals regardless of $\|W_0\|_F$ and $\|W_1\|_F$), and if we use an activation function like ReLU, $\|W_0\|_F$ does not have too much effect on $\|T^o_{1,aln}\|_F$ with $o \leq s$ (Fig. S1) since unlike $T^s_{1,sup}$, in $T^o_{1,aln}$, $f'\big(\cdot\big)$ participates instead of $f\big(\cdot\big)$.

In this respect, if $\|W_0\|_F$ is limited, $T^s_{1,sup}$ is less dominant in the equation

$$\Delta W_{1,FA}[k] \approx T^1_{1,aln}[k] + T^2_{1,aln}[k] + \cdots + T^s_{1,sup}[k] \tag{S2}$$

and the amount of alignment is expected to increase. Note that this analysis is more complicated in deep ANNs since $\|W_\ell\|_F$ affects $\|L_{\ell+1}\|_F$, and in this case, efficiency of network, in the sense of achieving more $\|L_{\ell+1}\|_F$ with less $\|W_\ell\|_F$, is important for achieving more alignment (more $\|T^o_{\ell,aln}\|_F, o \leq s$ and less $\|T^s_{\ell,sup}\|_F$).

In this regard, we applied proposed WN method and imposed some hypothetical constant random input matrix and variable error matrix with noisy autocorrelated elements on an open-loop nonlinear shallow network according to five cases below in order for investigating expansion of support of ACFs of error matrix elements from one step lag to two-step lag and also investigating effect of applying proposed WN method. In addition, we investigated the effect of these measures on the behavior of $T^1_{1,aln}$, $T^1_{1,sup}$ and summation of them from beginning to each current iteration ($\sum T^1_{1,aln}$ and $\sum T^1_{1,sup}$).

In all of the following cases, we set network dimensions $n_i = n_h = n_b = 100$, $n_o = 10$ and learning rate $\eta = 0.003$ and initialized W_0, W_1, B, X and E with i.i.d. elements from $\mathcal{N}(0, 1)$.

Case 1 In this case (Fig. S1A,B), we reinitialized E at every two iterations. As expected, $T^1_{1,aln}$ is partially aligned with the B^T in half of the iterations and $T^1_{1,sup}$ does not align. Consequently, $\sum T^1_{1,sup}$ does not provide any alignment and $\sum T^1_{1,aln}$ provides alignment even more than each individual $T^1_{1,aln}$ since there is a noise in $T^1_{1,aln}$ terms that cancels out in summation. The final amount of alignment of W_1 with B^T is the resultant of $\sum T^1_{1,sup}$ and $\sum T^1_{1,aln}$ while at each iteration, ΔW_1 does not provide any considerable amount of alignment, yet its accumulative effect on $W_1 \measuredangle B^T$ becomes apparent during iterations. In this case, Frobenius norms of feedforward weights continuously increase, which is so-called blow-up (Fig. 4B)

Case 2 In this case (Fig. S1C,D), case 1 is re-performed with the difference that we expanded the support of error matrix elements ACF from one-step to two-step by reinitializing E at every three iterations. The notable change compared to the case 1 is that in this case $\sum T^1_{1,sup}$ shows some amount of alignment, although it is subtle in each individual $T^1_{1,sup}$. In this case blow-up happens for weight matrices Frobenius norm (Fig. S1D).

Case 3 In this case (Fig. S1E,F) we repeated the previous case with the proposed WN method with $\gamma = 1$. This normalization results in more alignment compared to previous case. Although the $T^1_{1,aln}$ does not represent further alignment, the impact of normalization is notable on $T^1_{1,sup}$. In addition, ΔW_1 provides considerable amount of alignment which was subtle in the two previous cases.

Case 4 In this case (Fig. S1G,H), we re-performed the first case of this experiment with the proposed normalization with $\gamma = 1$. Compared with the first case ΔW_1 shows more alignment and $W_1 \measuredangle B_T$ almost reaches $T^1_{1,aln} \measuredangle B^T$ whereas $T^1_{1,sup}$ shows no alignment.

Case 5 To show the impact of weight matrices Frobenius norm on the final amount of alignment, In this case (Fig. S1I,J) we re-performed the fourth case of this experiment with the difference of $\gamma = 0.7$. Compared to the previous case $\|T^1_{1,sup}\|_F$ is reduced; thus, W_1 and ΔW_1 show slightly more alignment, yet the effect of reducing Frobenius norm of weight matrices finally saturates since this normalization method does not affect $T^1_{1,aln} \measuredangle B^T$.

In summary, expanding the support of ACFs of error matrix elements along lags and limiting Frobenius norm of forward weights, both improved alignment in this experiment.

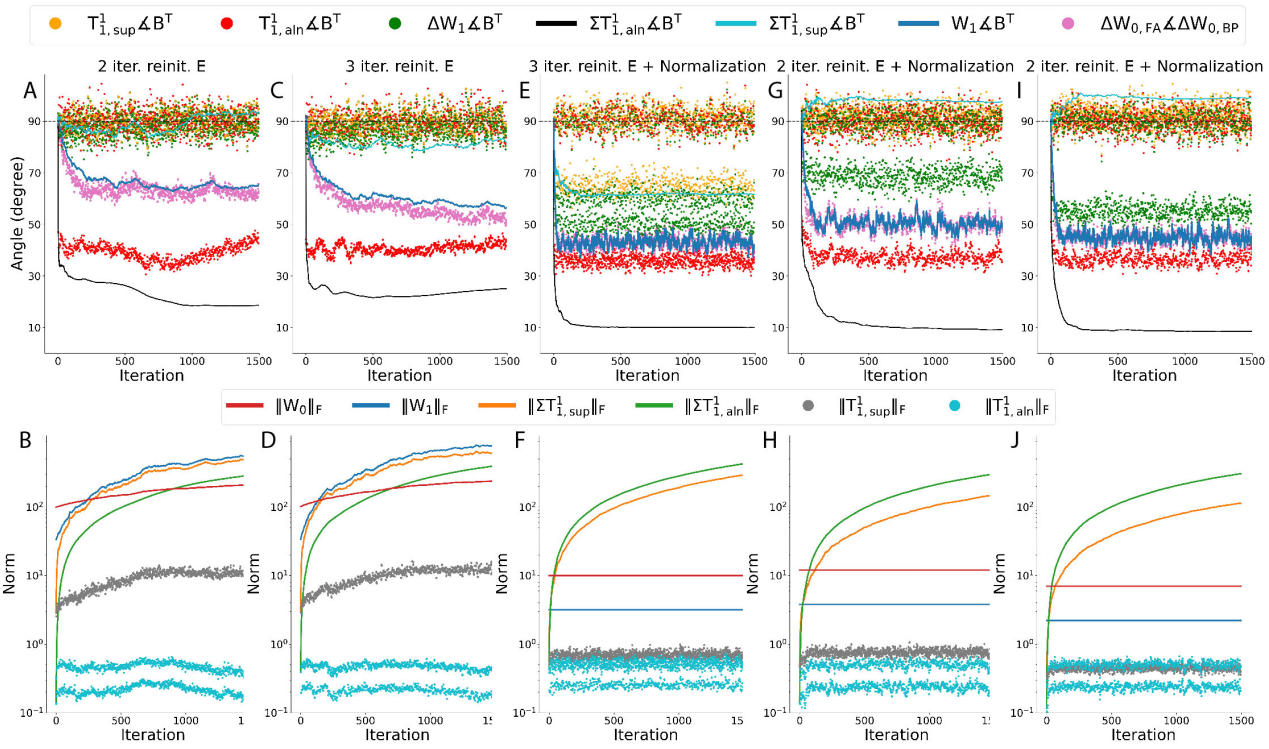

Figure S1 | Weight normalization and expanding support of ACF of error matrix elements affect the amount of alignment. Top row shows the behavior of $T^1_{1,aln}$, $T^1_{1,sup}$, the summation of them from the beginning to each current iteration ($\sum T^1_{1,aln}$ and $\sum T^1_{1,sup}$), and their effect on the final amount of alignment. Bottom row shows the behavior of Frobenius norm of alignment term of order 1, sup-alignment term of order 1, their summations and forward weight matrices.
(A,B) E is reinitialized at every two iterations and $T^1_{1,sup}$ does not align.
(C,D) E is reinitialized at every three iterations and $T^1_{1,sup}$ slightly aligns.
(B,D) Show Frobenius norm of forward weight matrices increase continuously and consequently $\|T^1_{1,sup}\|_F$ increases while $\|T^1_{1,aln}\|_F$ is relatively low and constant.
(E,F) E is reinitialized at every three iterations and normalization mechanism applied with $\gamma = 1$ hence $\|T^1_{1,sup}\|_F$ is limited and leads to more alignment.
(G,H) E is reinitialized at every two iterations and normalization mechanism applied with $\gamma = 1.2$ hence $\|T^1_{1,sup}\|_F$ is limited and leads to more alignment.
(I,J) The case of (G,H) is re-performed with the difference that $\gamma = 0.7$. Further reduction of Frobenius norm of forward weights leads to a slight increase of alignment. The effect of reducing Frobenius norm of forward weight matrices finally saturates since it does not increase the amount of alignment of $T^1_{1,aln}$.

Supplimentory Note 2. FA in an open-loop Linear Shallow ANN with constant error and input matrix

Consider a conventional two-layer linear neural network ($d = 2$ and $f(\cdot)$ is an identity function). Update directions computed by BP are

$$\Delta W_{1,BP}[k] = \eta L_1[k]^T E[k] \tag{S3}$$

$$\Delta W_{0,BP}[k] = \eta X[k]^T E[k] W_1[k]^T \tag{S4}$$

Update direction calculated by FA for W_1 is the same as BP (Equ. S4), but for updating W_0 error backpropagates by a constant random matrix B

$$\Delta W_{0,FA}[k] = \eta X[k]^T E[k] B \tag{S5}$$

Assume that FA learning process is performing in batch-mode and X is constant during iterations. Additionally, to give an initial intuition about FA, at first we assumed that E is constant and non-zero during iteration (this is not a realistic assumption in learning process). Indeed with this assumption, we intended to show that no learning process or decrease in loss function is required for alignment to happen and it occurs under certain conditions even if we break the feedback loop and feed the backward pass with a constant random E. According to these assumptions, after k iterations we have

$$W_0[k] = W_0[k-1] + \Delta W_{0,FA}[k-1] = W_0[0] + k\eta X^T E B \tag{S6}$$

and direction of W_0 converges to the direction of $X^T E B$

$$\lim_{k\to+\infty} \frac{W_0[k]}{\|W_0[k]\|_F} = \lim_{k\to+\infty} \frac{W_0[0] + k\eta X^T E B}{\|W_0[0] + k\eta X^T E B\|_F} = \frac{X^T E B}{\|X^T E B\|_F}. \tag{S7}$$

Therefore, for large enough k we have

$$W_0[k \gg 1] \simeq c_1 X^T E B \tag{S8}$$

where c_1 is a constant coefficient. Then

$$\Delta W_{1,FA}[k \gg 1] = \eta L_1^T E = \eta (X W_0)^T E \simeq \eta c_1 B^T E^T X X^T E. \tag{S9}$$

Like the convergence of W_0, for W_1 we have

$$W_1[k \gg 1] \simeq c_2 B^T E^T X X^T E = c_2 (E^T X X^T E B)^T. \tag{S10}$$

In order that $W_1[k \gg 1]$ aligns with B^T, we should have

$$W_1[k \gg 1] \measuredangle B^T \simeq cos^{-1}\left(\frac{\langle (E^T X X^T E B)^T, B^T \rangle_F}{\|B^T E^T X X^T E\|_F \|B\|_F}\right) < 90^\circ. \tag{S11}$$

Supplimentory Note 3. A semidefinite matrix as a transformation matrix

Consider $E \in \mathbb{R}^{n_b \times n_o}$, $X \in \mathbb{R}^{n_b \times n_i}$, and $B \in \mathbb{R}^{n_h \times n_o}$. The main factor for alignment of

$$W_1[k \gg 1] \simeq c_2 B^T E^T X X^T E = c_2 (E^T X X^T E B)^T. \tag{S12}$$

with B^T is $E^T X X^T E$ as a transformation matrix which is applied to B (or B^T depending on the notation in equation S12) and if it does not completely deviate B after transformation, alignment happens. Indeed it have some special properties that preserves the direction of B^T after matrix multiplication; otherwise, assuming A and B are two independent random matrix with i.i.d. elements from $\mathcal{N}(0,1)$ and proper dimensions, by statistical expectation, A totally deviates B and $AB \measuredangle A$ is 90° ($A, B \in \mathbb{R}^{100\times100}$, $SD = \pm 0.82^\circ$, one sample t-test, $p > 0.5, n = 1000$).

But $E^T X X^T E$ can be considered as the estimated autocorrelation matrix of data matrix $\sqrt{n_i} X^T E$ and has some special properties which leads to alignment. First, it is an $n_o \times n_o$ symmetric matrix and have n_o mutually orthogonal eigenvectors that totally span $\mathbb{R}^{n_o}$ [Strang et al., 1993]. Secondly, it is positive semidefinite [Strang et al., 1993]. It can be even positive definite if columns of $X^T E$ are linearly independent [Strang et al., 1993].

The impact of $E^T X X^T E$ as a transformation matrix on any arbitrary vector $\mathbf{v}$ after vector-matrix multiplication can be seen by decomposing $\mathbf{v}$ to the basis of orthonormal eigenvectors of $E^T X X^T E$. After multiplication, each of these components scales in its direction by a nonnegative eigenvalue (if it could be negative, the corresponding component would be able to totally change direction by 180°). Therefore this transformation does not totally deviate $\mathbf{v}$, yet the arrangement of its eigenvalues plays an essential role in the amount of deviation (Fig. 1D).

This analysis also holds in the case of matrix-matrix multiplication in equation S12 since transformation applies to every column of B independently. Columns of B are vectors in mutually exclusive subspaces of the whole space of vectorized matrix B. Therefore, if each column partially deviates in its own subspace, consequently so is the B.

Supplimentory Note 4. Sensitivity analysis

Learning process for handwritten digits classification is slightly sensitive to hyperparameters of network (width, η, γ and ...). 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 in different conditions ($\eta < 0.0005$ and $\eta < 0.0003$ and different γ) to analysis sensitivity of network with respect to networks parameters. In addition, in the following figures we used label smoothing (LS), by which for every 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.

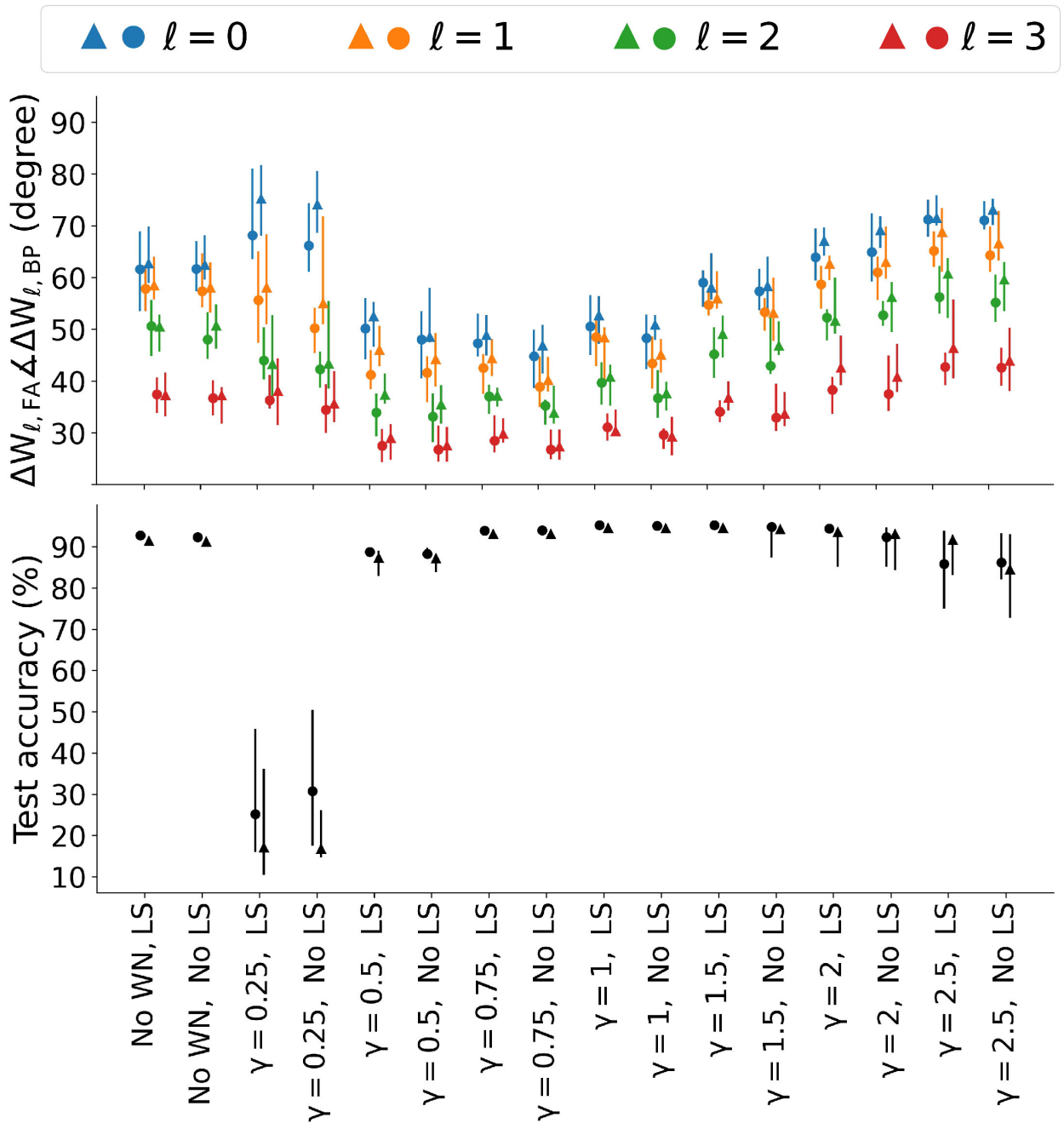

Figure S2 | Sensitivity analysis for final amount of alignment and final test accuracy. In training process of a five-layer ANN on MNIST dataset, final amount of alignment in different layers (upper panel) and final test accuracy (lower panel) is plotted in different conditions. Markers demonstrate median and error bars demonstrate minimum and maximum (n=10). Circle markers correspond to $\eta = 0.0005$ (learning rate) and triangle markers correspond to $\eta = 0.0003$. LS: label smoothing, WN: weight normalization

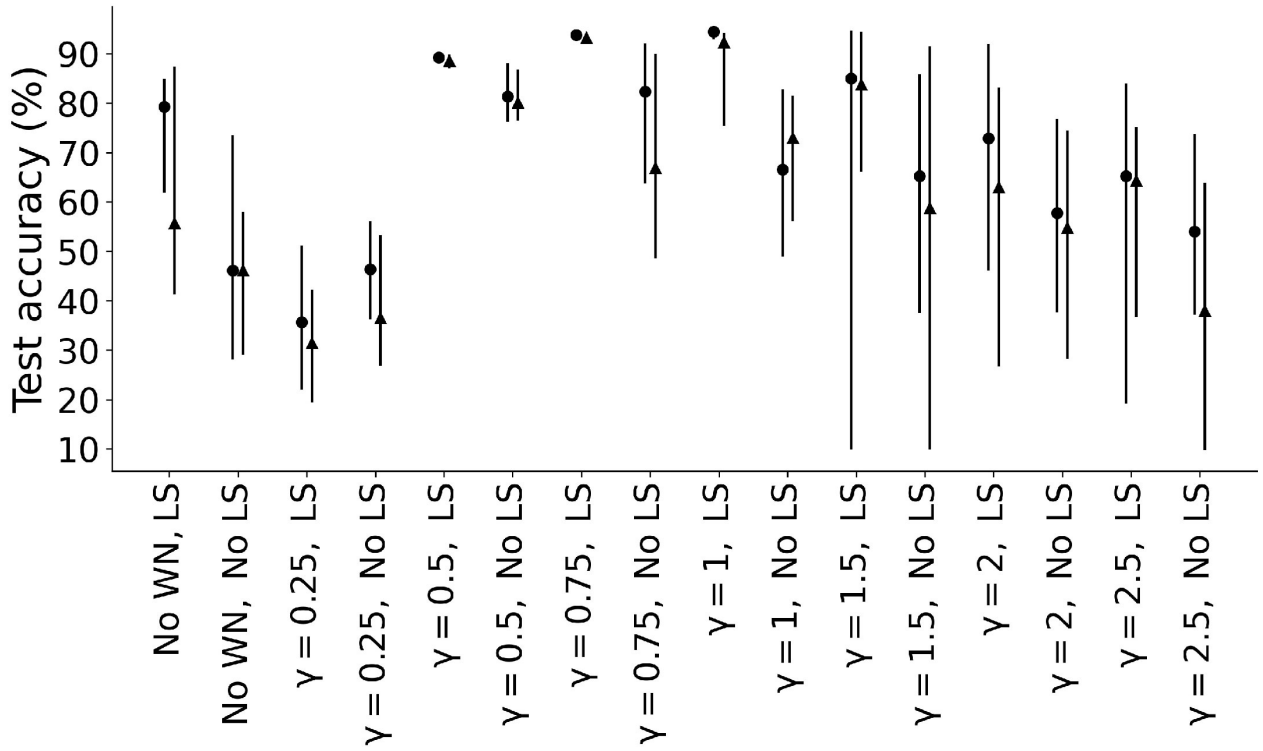

Figure S3 | Sensitivity analysis for final test accuracy after label changing. In training process of a five-layer ANN on MNIST dataset, where true labels of data is changed at epoch 100, final amount of test accuracy at epoch 200 is plotted in different conditions. Markers demonstrate median and error bars demonstrate minimum and maximum (n=10). 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

[Strang et al., 1993] Strang, G., Strang, G., Strang, G., and Strang, G. (1993). *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA.