A PSEUDOCODE FOR PEPITA

Algorithm S1 describes the original PEPITA as presented in Dellaferrera & Kreiman (2022). Algorithm S2 describes our modification of PEPITA, which is Hebbian and local both in space and in time (Pepita-time-local).

Algorithm S1 Implementation of PEPITA

Require: Input x and one-hot encoded label y
  {standard forward pass}
  $h_0 = x$
  for $\ell = 1, ..., L$ **do**
    $h_\ell = \sigma_\ell(W_\ell h_{\ell-1})$
  end for
  $e = h_L - y$
  {modulated forward pass}
  $h_0^{err} = x + Fe$
  for $\ell = 1, ..., L$ **do**
    $h_\ell^{err} = \sigma_\ell(W_\ell h_{\ell-1}^{err})$
    if $\ell < L$ **then**
      $\Delta W_\ell = (h_\ell - h_\ell^{err})(h_{\ell-1}^{err})^\top$
    else
      $\Delta W_\ell = e(h_{\ell-1}^{err})^\top$
    end if
    $W_\ell(t+1) = W_\ell(t) - \eta \Delta W_\ell$ {apply update}
  end for

Algorithm S2 Impl. of PEPITA-time-local

Require: Input x and one-hot encoded label y
  {standard forward pass}
  $h_0 = x$
  for $\ell = 1, ..., L$ **do**
    $h_\ell = \sigma_\ell(W_\ell h_{\ell-1})$
    $\Delta W_\ell^+ = h_\ell h_{\ell-1}^\top$
    $W_\ell^+(t+1) = W_\ell(t) - \eta \Delta W_\ell^+$ {apply 1st update}
  end for
  $e = h_L - y$
  {modulated forward pass}
  $h_0^{err} = x - Fe$
  for $\ell = 1, ..., L$ **do**
    $h_\ell^{err} = \sigma_\ell(W_\ell^+ h_{\ell-1}^{err})$
    if $\ell < L$ **then**
      $\Delta W_\ell^- = -h_\ell^{err} h_{\ell-1}^{err\top}$
    else
      $\Delta W_\ell^- = -y h_{\ell-1}^{err\top}$
    end if
    $W_\ell(t+1) = W_\ell^+(t+1) - \eta \Delta W_\ell^-$ {apply 2nd update}
  end for

The updates for PEPITA-Hebbian are:

- for the hidden layers:

$$\begin{aligned} \Delta W_\ell &= h_\ell h_{\ell-1}^{err\top} - h_\ell^{err} h_{\ell-1}^{err\top} \\ &\simeq h_\ell h_{\ell-1}^\top - h_\ell^{err} h_{\ell-1}^{err\top}, \end{aligned} \tag{11}$$

- for the first and last layers

$$\begin{aligned} \Delta W_1 &\simeq h_1 x^\top - h_1^{err}(x - Fe)^\top; \\ \Delta W_L &\simeq h_L h_{L-1}^\top - y h_{L-1}^{err\top}. \end{aligned} \tag{12}$$

These updates are applied at the end of both forward passes for PEPITA-Hebbian, similarly as in pseudocode S1.

B TRAINING WITH THE TIME-LOCAL RULE

The weight update of the *PEPITA-Hebbian* rule in eqn. 8 consists of two separate phases that can contribute to learning without knowledge of the activity of the other, i.e. learning is time-local. Specifically, the term $h_\ell h_{\ell-1}^\top$ can be applied *online*, immediately as the activations of the first pass are computed. Analogously, the second term $-h_\ell^{err} h_{\ell-1}^{err\top}$ can be applied immediately during the second forward pass. To ensure that the hidden-layer updates prescribed by PEPITA are useful, we compared the test curve of PEPITA-TL against a control with $F = 0$ (Fig. S1), and found that removing the feedback decreases the accuracy from approx. 40% to approx. 18%.

Regarding the time-locality of FF, the two forward passes can be computed in parallel, as the *modulated* pass does not need to wait for the computation of the error of the first pass. However, according to the available implementations (Mukherjee, 2023) the updates related to both passes are applied together at the end of the second forward pass.

The time-local PEPITA was trained on the CIFAR-10 dataset with learning rate 0.01. All the other hyperparameters are the same as the ones reported in Table S3.

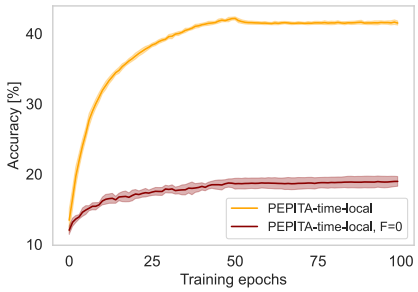

Figure S1: Test curve for PEPITA in its time-local formulation and time-local PEPITA with F=0 (i.e., only the last layer is trained) on the CIFAR-10 dataset. The network has 1 hidden layer with 1024 units. The forward matrices are initialized using the He normal initialization. F entries are sampled from a normal distribution with standard deviation $0.5{\cdot}2\sqrt{6/(32 \cdot 32 \cdot 3)}$. We use learning rate 0.0001 and weight decay with $\lambda = 10^{-4}$. The learning is reduced by a factor of $\times 0.1$ at epoch 50. The plot indicates mean and standard deviation over 10 independent runs. Time-local PEPITA achieves a significantly higher accuracy than the time-local, F=0 scheme.

C DISTRIBUTION OF THE GOODNESS IN PEPITA

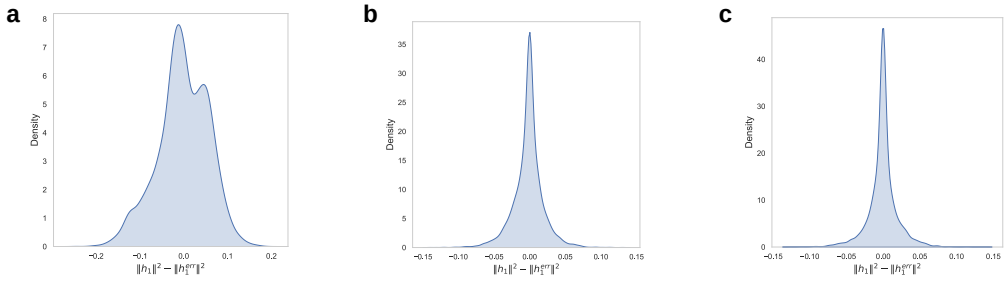

Figure S2: Difference of the norm of the squared activities of the first hidden layer between the *clean* and *modulated* pass in PEPITA (a) before training, (b) after 50 epochs, and (c) at the end of training. The network is a 2-hidden-layer network trained with WD with $\lambda = 10^{-4}$ on the CIFAR-10 dataset. The activites are recorded on the test set. We remark that in PEPITA the input of the second forward pass is modulated by the error. Since the error decreases during training, also the difference of the activations in the two passes decreases with training. This explains why the distribution of the difference of the norm of the squared activities has a lower standard deviation in the middle of training (b) and at the end of training (c), than before training (a). In contrast, the modulation of the input in FF is constant during training, and the scope of training is maximising the difference of the goodness in the two passes.

D ADDITIONAL FIGURES ON THE AFA APPROXIMATION

Fig. S3 displays the comparison between the "vanilla" PEPITA algorithm and the AFA approximation introduced in eqn. 6 of the main text. The test accuracy as a function of training epochs is depicted for the CIFAR-10 dataset in the left panel and for the CIFAR-100 dataset in the right panel.

Fig. S4 depicts the norm of the adaptive feedback matrix $f = W_1 F/D$ as a function of training time. The symbols mark the numerical simulations at dimension $D = 500$, while the full line represents our theoretical prediction. We observe that, for this run, the norm of the adaptive feedback increases over time. We have observed by numerical inspection that this behavior is crucial to speed up the dynamics, as also observed in Dellaferrera & Kreiman (2022).

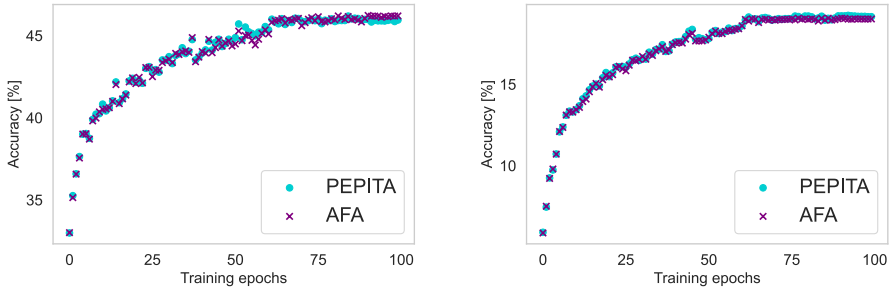

Figure S3: Comparison between the test accuracy as a function of training epochs between the "vanilla" PEPITA algorithm and its AFA approximation (eqn. 6 of the main text) for the CIFAR-10 (*left panel*) and CIFAR-100 (*right panel*) datasets.

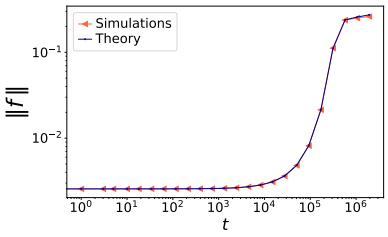

Figure S4: Norm of the alignment matrix $f = W_1 F/D$ as a function of training time, for the same parameters as in Fig. 2 of the main text: $D = 500$, $lr = .05$, erf activation, two hidden units in both teacher and student ($K = M = 2$).

E ODES FOR ONLINE LEARNING IN THE TEACHER-STUDENT REGRESSION TASK

In this section, we present the details of the teacher-student model under consideration and we sketch the derivation of the ordinary differential equations (ODEs) tracking the online learning dynamics of the AFA rule. We consider a shallow *student* network trained with AFA to solve a supervised learning task. The input data are random $D-$dimensional vectors $x \in \mathbb{R}^D$ with independent identically distributed (i.i.d.) standard Gaussian entries $x_j \sim \mathcal{N}(0,1)$, $j = 1, \dots, D$, and the (scalar) labels are generated as the output of a 1-hidden-layer *teacher* network with parameters $\tilde{\theta} = (\tilde{W}_1, \tilde{W}_2, M, \tilde{\sigma})$:

$$y = \sum_{m=1}^{M} \tilde{W}_2^m \tilde{\sigma}(\nu^m), \qquad \nu^m = \frac{\tilde{W}_1^m x}{\sqrt{D}}, \tag{13}$$

where M is the size of the teacher hidden layer, ν^m denotes the teacher preactivation at unit $m \in \{1, \dots, M\}$, and $\tilde{\sigma}(\cdot)$ is the activation function. The student is a 1-hidden-layer neural network

parametrized by $\theta = (W_1, W_2, K, \sigma)$ that outputs the prediction

$$\hat{y} = \sum_{k=1}^{K} W_2^k \sigma(\lambda^k), \qquad \lambda^k = \frac{W_1^k x}{\sqrt{D}}, \tag{14}$$

where K is the size of the student hidden layer, $\sigma(\cdot)$ the student activation function, λ^k the student preactivation at unit $k \in \{1, \dots, K\}$. For future convenience, we write explicitly the scaling with respect to the input dimension. Therefore, at variance with the main text, in this supplemementary section we always rescale the first layer weights as well as the feedback by $1/\sqrt{D}$.

We focus on the *online* (or *one-pass*) learning protocol, so that at each training time the student network is presented with a fresh example x_μ, $\mu = 1, \dots N$, and $N/D \sim \mathcal{O}_D(1)$. The weights are updated according to the Adaptive Feedback Alignment (AFA) rule defined in eqn. 6:

$$W_1(\mu+1) = W_1(\mu) - \eta_1 \Delta W_1(\mu), \qquad \Delta W_1 = \frac{W_1 F}{D} e\, h_1' \frac{x^\top}{\sqrt{D}}, \tag{15}$$

$$W_2(\mu+1) = W_2(\mu) - \eta_2 \Delta W_2(\mu), \qquad \Delta W_2 = e\, h_1^\top. \tag{16}$$

We consider fixed learning rates $\eta_1 = \eta$, $\eta_2 = \eta/D$. Different learning rate regimes have been explored in Veiga et al. (2022). It is crucial to notice that the mean squared generalization error

$$\epsilon_g(\theta, \tilde{\theta}) = \frac{1}{2}\mathbb{E}_x \left[\left(\sum_{k=1}^{K} W_2^k \sigma(\lambda^k) - \sum_{m=1}^{M} \tilde{W}_2^m \tilde{\sigma}(\nu^m) \right)^2 \right] \tag{17}$$

depends on the high-dimensional input expectation only through the low-dimensional expectation over the preactivations $\{\lambda_k\}_{k=1}^K, \{\nu_m\}_{m=1}^M$. Notice that, in this online-learning setup, the input x is independent of the weights, which are held fixed when taking the expectation. Furthermore, due to the Gaussianity of the inputs, the preactivations are also jointly Gaussian with zero mean and second moments:

$$Q^{kl} = \mathbb{E}_x \left[\lambda^k \lambda^l\right] = \frac{W_1^k \cdot W_1^l}{D}\,, \tag{18}$$

$$R^{km} = \mathbb{E}_x \left[\lambda^k \nu^m\right] = \frac{W_1^k \cdot \tilde{W}_1^m}{D}\,, \tag{19}$$

$$T^{mn} = \mathbb{E}_x \left[\nu^m \nu^n\right] = \frac{\tilde{W}_1^m \cdot \tilde{W}_1^n}{D}. \tag{20}$$

The above matrices are named *order parameters* in the statistical physics literature and play an important role in the interpretation. The matrices Q and T capture the self-overlap of the student and teacher networks respectively, while the matrix R encodes the teacher-student overlap. In the infinite-dimensional limit discussed above, the generalization error is only a function of the order parameters Q, T, R and of the second layer weights $\tilde{W}_2, W_2$ of teacher and student respectively. Therefore, by tracking the evolution of these matrices via a set of ODEs – their "equations of motion" – we obtain theoretical predictions for the learning curves. The update equations for Q, R, W_2 can be obtained from eqn. 15 according to the following rationale. As an example, we consider the update equation for the matrix Q:

$$\begin{aligned} &Q^{kl}(\mu+1) - Q^{kl}(\mu) \\ &\quad = \frac{1}{D}\left[W_1^k(\mu) - \eta \Delta W_1^k(\mu)\right] \cdot \left[W_1^l(\mu) - \eta \Delta W_1^l(\mu)\right] - \frac{1}{D} W_1^k(\mu) \cdot W_1^l(\mu) \\ &\quad = -\frac{1}{D}\eta\, f^k\, e\, \sigma'(\lambda^k)\lambda^l - \frac{1}{D}\eta\, f^l\, e\, \sigma'(\lambda^l)\lambda^k + \frac{1}{D}\eta^2\, f^k\, f^l\, \sigma'(\lambda^k)\sigma'(\lambda^l)e^2, \end{aligned} \tag{21}$$

where we have defined the adaptive feedback $f := W_1 F/D$, we have used that $\|x_\mu\|^2 = D$ as $D \to \infty$ and omitted the $\mu-$dependence on the right hand side for simplicity. By taking $t = \mu/D$, as shown in Goldt et al. (2019), in the infinite-dimensional limit $Q^{kl}(\mu)$ concentrates to the solution of the following ODE:

$$\frac{\mathrm{d}Q^{kl}}{\mathrm{d}t} = -\eta f^k \mathbb{E}\left[\sigma'(\lambda^k)\lambda^l e\right] - \eta f^l \mathbb{E}\left[\sigma'(\lambda^l)\lambda^k e\right] + \eta^2 f^k f^l \mathbb{E}\left[\sigma'(\lambda^k)\sigma'(\lambda^l)e^2\right], \tag{22}$$

Similarly, we can derive ODEs for the evolution of R, W_2 and the adaptive feedback f:

$$\frac{\mathrm{d}R^{km}}{\mathrm{d}t} = -\eta\, f^k \mathbb{E}\left[\sigma'(\lambda^k)\nu^m e\right], \quad \frac{\mathrm{d}W_2^k}{\mathrm{d}t} = -\eta \mathbb{E}\left[\sigma(\lambda^k)e\right], \quad \frac{\mathrm{d}f^k}{\mathrm{d}t} = -\eta f^k \mathbb{E}\left[\rho\sigma'(\lambda^k)e\right], \tag{23}$$

where the expectations are taken over the preactivatons and $\rho = F\,x/\sqrt{D}$, and we have $\mathbb{E}_x[\lambda^k\rho] = f^k$, $\mathbb{E}_x[\nu^m\rho] = \tilde{f}^m := \tilde{W}_1 F/D$, $q_f := F \cdot F/D$. The generalization error can be rewritten as

$$\lim_{D\to\infty} \epsilon_g(\theta,\tilde{\theta}) = \frac{1}{2}\sum_{k,l=1}^{K} W_2^k W_2^l\, I_2(k,l) + \frac{1}{2}\sum_{m,n=1}^{M} \tilde{W}_2^m \tilde{W}_2^n\, I_2(m,n) - \sum_{k=1}^{K}\sum_{m=1}^{M} W_2^k \tilde{W}_2^m\, I_2(k,m), \tag{24}$$

where I_2 generically encodes the averages over the activations

$$I_2(\alpha,\beta) = \mathbb{E}\left[\sigma_\alpha(\gamma^\alpha)\sigma_\beta(\gamma^\beta)\right], \quad \gamma^\alpha = \begin{cases}\lambda^k & \text{if } \alpha = k,l \\ \nu^m & \text{if } \alpha = m,n\end{cases}, \quad \sigma_\alpha = \begin{cases}\sigma & \text{if } \alpha = k,l \\ \tilde{\sigma} & \text{if } \alpha = m,n\end{cases}. \tag{25}$$

The other averages in eqns. can be expressed in a similar way and estimated by Monte Carlo methods. In the case of sigmoidal activation $\sigma(x) = \mathrm{erf}(x/\sqrt{2})$, the function I_2 has an analytic expression.

$$I_2(\alpha,\beta) = \frac{2}{\pi}\mathrm{arcsin}\frac{C^{\alpha\beta}}{\sqrt{1+C^{\alpha\alpha}}\sqrt{1+C^{\beta\beta}}}, \quad C^{kl} = Q^{kl}, \;\; C^{km} = R^{km}, \;\; C^{mn} = T^{mn}. \tag{26}$$

E.1 Early-training expansion

As done by Refinetti et al. (2021a) for DFA, it is instructive to consider an expansion of the ODEs at early training times. We assume the following initialization: $W_2^k(0) = 0$, $\forall k \in \{1,\dots,K\}$, while the first layer is assumed to be orthogonal to the teacher $W_1^k(0)\cdot\tilde{W}_1^m = 0$ and of fixed norm $\|W_1^k(0)\|^2/D = q_0$, $\forall k \in \{1,\dots,K\}, \forall m \in \{1,\dots,M\}$. We also take orthogonal first-layer teacher weights, such that T is the identity matrix. This initialization leads to:

$$R^{km}(0) = 0\,, \qquad \frac{\mathrm{d}}{\mathrm{d}t}W_2^k\Big|_{t=0} = 0\,, \qquad \frac{\mathrm{d}}{\mathrm{d}t}R^{km}\Big|_{t=0} = \frac{\sqrt{2}}{\pi\sqrt{1+q_0}}\eta\, f^k(0)\,\tilde{W}_2^m. \tag{27}$$

We can therefore compute the second-layer update to linear order:

$$\frac{\mathrm{d}}{\mathrm{d}t}W_2^k(t) = \frac{2}{\pi^2(1+q_0)}\eta^2\, f^k(0)\,\|\tilde{W}_2\|_2^2\; t + \mathcal{O}(t^2). \tag{28}$$

Eqn. 28 shows that the update of the second-layer weights at early training times is in the direction of the adaptive feedback matrix, in agreement with the alignment phase observed in experiments. Crucially, it is necessary that $f^k(0) \neq 0\ \forall k$ in order to have non-zero updates, i.e. the feedback F must not be orthogonal to the first-layer weights at initialization. We now inspect the behavior of the adaptive feedback at the beginning of training. We have that the update at time zero is:

$$\frac{\mathrm{d}}{\mathrm{d}t}f^k\Big|_{t=0} = \frac{2}{\pi}\,\eta\frac{f^k(0)}{\sqrt{(1+q_0)(1+q_f) - f^k(0)^2}}\,(\tilde{W}_2\cdot\tilde{f}), \tag{29}$$

and

$$f^k(t) = f^k(0) + \frac{2}{\pi}\,\eta\frac{f^k(0)}{\sqrt{(1+q_0)(1+q_f) - f^k(0)^2}}\,(\tilde{W}_2\cdot\tilde{f})\; t + \mathcal{O}(t^2). \tag{30}$$

Eqn. 29 illustrates that the feedback-teacher alignment $\tilde{f}$ plays an important role in speeding up the dynamics. Indeed, if $\|\tilde{f}\|_2^2$ is close to zero, the feedback update slows down inducing long plateaus in the generalization error. A similar role is played by the alignment angle between $\tilde{W}_2$ and $\tilde{f}$.

F PEPITA'S RESULTS COMPARED TO THE BASELINES

Table S1: Test accuracy [%] achieved by BP, FA, DRTP, PEPITA, and PEPITA with WM in the experiments. Mean and standard deviation are computed over 10 independent runs. The nonlinearity is ReLU for all algorithms except DRTP, for which is tanh. WM was used in combination with weight decay with $\lambda = 10^{-4}$ for the networks trained on the CIFAR-10 dataset and for the 3-hidden-layer networks trained on the CIFAR-100 dataset. The other networks are trained without weight decay. Bold fonts refer to the best results exclusively among PEPITA and its improvements.

	2 HIDDEN LAYERS			3 HIDDEN LAYERS		
	MNIST	CIFAR-10	CIFAR-100	MNIST	CIFAR-10	CIFAR-100
BP	98.85±0.06	59.69±0.25	32.28±0.17	98.89±0.04	60.07±0.28	32.80±0.16
FA	98.64±0.05	57.76±0.39	22.90±0.14	97.48±0.06	52.99±0.21	22.81±0.21
DRTP	95.36±0.09	47.48±0.19	20.55±0.30	95.74±0.10	47.44±0.19	21.81±0.24
PEPITA	**98.19**±0.07	52.39±0.27	24.88±0.15	95.07±0.11	52.47±0.24	01.00±0.00
PEPITA +WD	98.09±0.07	53.09±0.33	24.64±0.24	95.09±0.16	52.56±0.31	**23.13**±0.21
PEPITA +WM	98.13±0.05	**53.44**±0.28	**26.95**±0.24	**96.33**±0.12	**52.80**±0.33	23.03±0.28

G SLOWNESS RESULTS

In our analysis of convergence rate, we used the *plateau equation for learning curves* (Dellaferrera et al., 2022):

$$\text{accuracy} = \frac{\text{max\_accuracy} \cdot \text{epochs}}{\text{slowness} + \text{epochs}} \tag{31}$$

By fitting the test curve to this equation, we calculated the slowness parameter, which measures how quickly the network reduces error during training. In mathematical terms, the slowness value corresponds to the number of epochs needed to reach half of the maximum accuracy. In practice, a lower slowness value indicates faster training.

Table S2 shows that the best convergence rate for PEPITA (*i.e.*, small slowness value) is obtained in general by PEPITA with WM (MNIST, CIFAR-100). Compared to the baselines, PEPITA has a better convergence than all the algorithms on MNIST, is the slowest on CIFAR-10, and the second best after BP on CIFAR-100. However, these results are strongly dependent on the chosen learning rate.

Table S2: Convergence rate in terms of slowness value obtained by BP, FA, DRTP and PEPITA in the experiments for the fully connected models trained on MNIST, CIFAR-10 and CIFAR-100 (same simulations reported in Table 2). PreM refers to pre-mirroring (Sec. 4). The smallest the slowness value, the better the convergence rate. The slowness is computed on the first 60 epochs of the test curve (before the learning rate decay), averaged over 10 independent runs. All the networks are trained without weight decay. The slowness result of FA on CIFAR-10 is lower than in Dellaferrera & Kreiman (2022) as our grid search returned a higher value for the learning rate. Bold fonts refer to the best results exclusively among PEPITA and its improvements.

	1×1024 FULLY CONNECTED MODELS		
	MNIST	CIFAR-10	CIFAR-100
BP	0.061±0.001	0.421±0.016	1.406±0.053
FA	0.081±0.002	0.463±0.020	4.946±0.123
DRTP	0.059±0.002	0.362±0.021	12.904±0.443
PEPITA	0.052±0.004	0.894±0.071	2.695±0.166
PEPITA+WM	0.047±0.005	0.890±0.081	2.333±0.102
PEPITA+WM+PREM	**0.040**±0.002	**0.856**±0.051	**1.999**±0.059

H HYPERPARAMETERS

Table S3: 1-hidden-layer network architectures and settings used in the experiments. The nonlinearity is ReLU for all algorithms except DRTP, for which is tanh.

	1 HIDDEN LAYER			1 HIDDEN LAYER - NORMALIZATION		
	MNIST	CIFAR-10	CIFAR-100	MNIST	CIFAR-10	CIFAR-100
INPUTSIZE	28×28×1	32×32×3	32×32×3	28×28×1	32×32×3	32×32×3
HIDDEN UNITS	1×1024	1×1024	1×1024	1×1024	1×1024	1×1024
OUTPUT UNITS	10	10	100	10	10	100
η BP	0.1	0.01	0.1	–	–	–
η FA	0.1	0.01	0.01	–	–	–
η DRTP	0.01	0.001	0.001	–	–	–
η PEPITA	0.1	0.01	0.01	100	10	100
λ WEIGHT DECAY	10^{-5}	10^{-4}	10^{-5}	0.0	0.0	0.0
η DECAY RATE	×0.1	×0.1	×0.1	×0.5	×0.1	×0.1
DECAY EPOCH	60,90	60,90	60,90	60,90	60,90	60,90
BATCH SIZE	64	64	64	64	64	64
η WM	0.1	0.001	0.1	0.001	0.001	0.001
λ WEIGHT DECAY WM	0.0	0.1	0.5	0.1	0.1	0.1
$\sigma_F^{(0)}$ (UNIFORM)	$0.05{\cdot}2\sqrt{\frac{6}{\text{FANIN}}}$	$0.05{\cdot}2\sqrt{\frac{6}{\text{FANIN}}}$	$0.05{\cdot}2\sqrt{\frac{6}{\text{FANIN}}}$	$0.05{\cdot}2\sqrt{\frac{6}{\text{FANIN}}}$	$0.05{\cdot}2\sqrt{\frac{6}{\text{FANIN}}}$	$0.05{\cdot}2\sqrt{\frac{6}{\text{FANIN}}}$
$\sigma_F^{(0)}$ (NORMAL)	$0.05{\cdot}\sqrt{\frac{2}{\text{FANIN}}}$	$0.05{\cdot}\sqrt{\frac{2}{\text{FANIN}}}$	$0.05{\cdot}\sqrt{\frac{2}{\text{FANIN}}}$	$0.05{\cdot}\sqrt{\frac{2}{\text{FANIN}}}$	$0.05{\cdot}\sqrt{\frac{2}{\text{FANIN}}}$	$0.05{\cdot}\sqrt{\frac{2}{\text{FANIN}}}$
FAN IN	$28 \cdot 28 \cdot 1$	$32 \cdot 32 \cdot 3$	$32 \cdot 32 \cdot 3$	$28 \cdot 28 \cdot 1$	$32 \cdot 32 \cdot 3$	$32 \cdot 32 \cdot 3$
#EPOCHS	100	100	100	100	100	100
DROPOUT	10%	10%	10%	10%	10%	10%

Table S4: 2-, 3-hidden-layer network architectures and settings used in the experiments. The nonlinearity is ReLU for all algorithms except DRTP, for which is tanh. **(*)** For the 3-hidden-layer network trained with PEPITA on the MNIST dataset, we did not use learning rate decay, as indicated by the grid search.

	2 HIDDEN LAYERS			3 HIDDEN LAYERS		
	MNIST	CIFAR-10	CIFAR-100	MNIST	CIFAR-10	CIFAR-100
INPUTSIZE	28×28×1	32×32×3	32×32×3	28×28×1	32×32×3	32×32×3
HIDDEN UNITS	2×1024	2×1024	2×1024	3×1024	3×1024	3×1024
OUTPUT UNITS	10	10	100	10	10	100
η BP	0.1	0.01	0.1	0.1	0.01	0.1
η FA	0.1	0.01	0.01	0.01	0.001	0.01
η DRTP	0.001	0.001	0.001	0.001	0.001	0.001
η PEPITA	0.1	0.01	0.01	0.001	0.01	0.01
λ WEIGHT DECAY	10^{-5}	10^{-4}	10^{-5}	10^{-5}	10^{-4}	10^{-4}
η DECAY RATE (*)	$\times 0.1$	$\times 0.1$	$\times 0.1$	$\times 0.1$	$\times 0.1$	$\times 0.1$
DECAY EPOCH	60,90	60,90	60,90	60,90	60,90	60,90
BATCH SIZE	64	64	64	64	64	64
η WM	0.00001	1.0	1.0	0.1	0.001	0.001
λ WD WM	0.0	0.1	0.1	0.001	0.1	0.1
$\sigma_F^{(0)}$ (UNIFORM)	$0.05 \cdot 2\sqrt{\frac{6}{\text{FANIN}}}$	$0.05 \cdot 2\sqrt{\frac{6}{\text{FANIN}}}$	$0.05 \cdot 2\sqrt{\frac{6}{\text{FANIN}}}$	$0.05 \cdot 2\sqrt{\frac{6}{\text{FANIN}}}$	$0.05 \cdot 2\sqrt{\frac{6}{\text{FANIN}}}$	$0.05 \cdot 2\sqrt{\frac{6}{\text{FANIN}}}$
$\sigma_F^{(0)}$ (NORMAL)	$0.05 \cdot \sqrt{\frac{2}{\text{FANIN}}}$	$0.05 \cdot \sqrt{\frac{2}{\text{FANIN}}}$	$0.05 \cdot \sqrt{\frac{2}{\text{FANIN}}}$	$0.05 \cdot \sqrt{\frac{2}{\text{FANIN}}}$	$0.05 \cdot \sqrt{\frac{2}{\text{FANIN}}}$	$0.05 \cdot \sqrt{\frac{2}{\text{FANIN}}}$
FAN IN	$28 \cdot 28 \cdot 1$	$32 \cdot 32 \cdot 3$	$32 \cdot 32 \cdot 3$	$28 \cdot 28 \cdot 1$	$32 \cdot 32 \cdot 3$	$32 \cdot 32 \cdot 3$
#EPOCHS	100	100	100	100	100	100
DROPOUT	10%	10%	10%	10%	10%	10%

I TRAINING DEEPER FULLY-CONNECTED MODELS

In the field of biologically inspired learning, reaching convergence on networks with more than a couple of hidden layers is a long standing challenge. Two significant reasons for this are the difficulty of learning hierarchical representations and the explosion of weight updates and activities in deep layers Illing et al. (2021). In particular, forward-only learning schemes have been shown to work so far on a maximum of four hidden layers (FF (Hinton, 2022)). The original PEPITA paper was only able to train 1 hidden layer models (Dellaferrera & Kreiman, 2022), therefore our strategy to reach convergence with up to 5 hidden layers (Fig. S5) represents a significant improvement over the previous work.

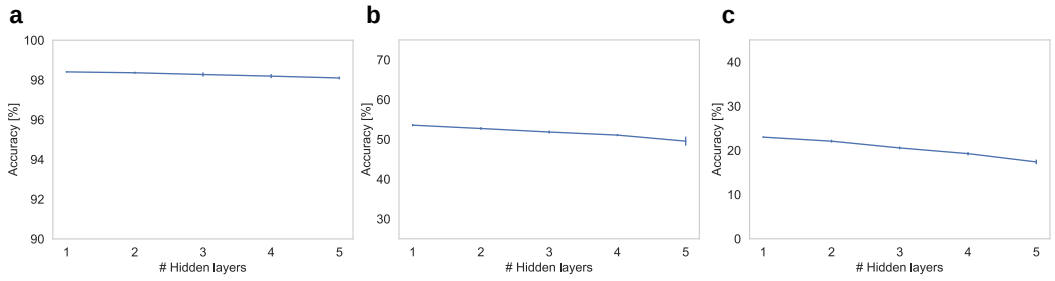

Figure S5: Test accuracy obtained with PEPITA and normalization of the activations for 1- to 5-hidden-layer networks on (a) MNIST, (b) CIFAR-10, and (c) CIFAR-100. Note that compared to Fig. 3 the trend is decreasing, as here we used activation normalization to obtain convergence for 4 and 5 hidden layers, which we did not use for Fig. 3.

J WEIGHT MIRRORING

Algorithm S3 describes how WM is applied to PEPITA. This algorithm is applied at the end of each training epoch. $\sigma_F^{(t+1)}$ refers to the standard deviation of the entries of $F(t+1) = \prod_{\ell=1}^{L} F_\ell(t+1)$.

Algorithm S3 Implementation of WM

{Mirror weights}
for $\ell = 1, ..., L$ **do**
  $\delta_{\ell-1} \sim \mathcal{N}(\mu, \sigma^2)$
  $\delta_\ell = \sigma_\ell(W_\ell \delta_{\ell-1})$
  $F_\ell(t+1) = F_\ell(t) - \eta \delta_{\ell-1} \delta_\ell^\top$
end for
{Normalize feedback matrices}
for $\ell = 1, ..., L$ **do**
  $F_\ell(t+1) = \left(\sigma_F^{(0)} / \sigma_F^{(t+1)}\right)^{1/L} \cdot F_\ell$
end for

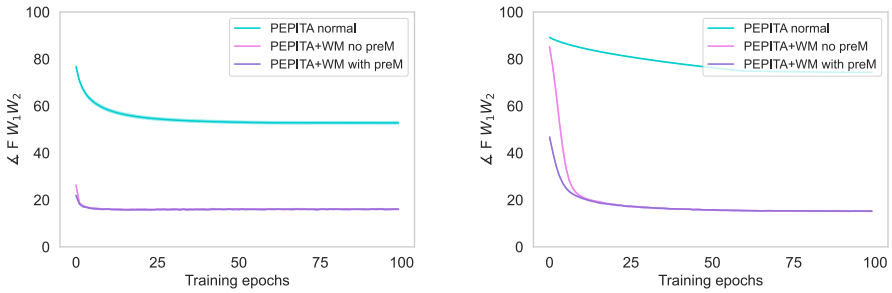

Figure S6: Alignment angle between F and W_{tot} during training with (pink and purple curves) or without (blue curve) WM for the MNIST (*left panel*) and CIFAR-100 (*right panel*) datasets. PreM refers to pre-mirroring (Sec. 4). The hyperparameters are reported in Table S3. The plots indicate mean and standard deviation over 10 independent runs.

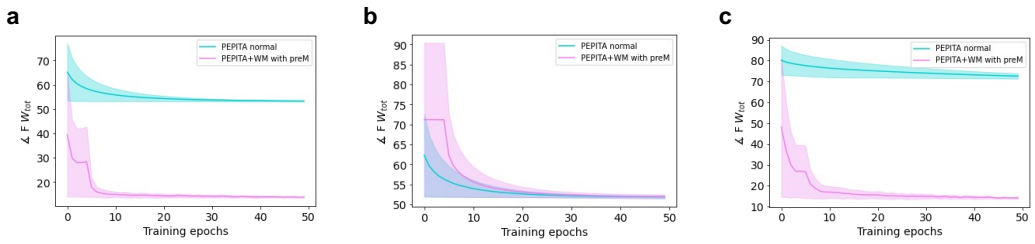

Figure S7: Alignment angle between F and W_{tot} during training with (pink curve) or without (blue curve) WM for the MNIST dataset for (a) 1-, (b) 2-, (c) 3-hidden-layer networks. The hyperparameters are reported in Table S3 for the 1-hidden-layer network and Table S4 for the 2- and 3-hidden-layer networks. The plots indicate mean and standard deviation over 10 independent runs.

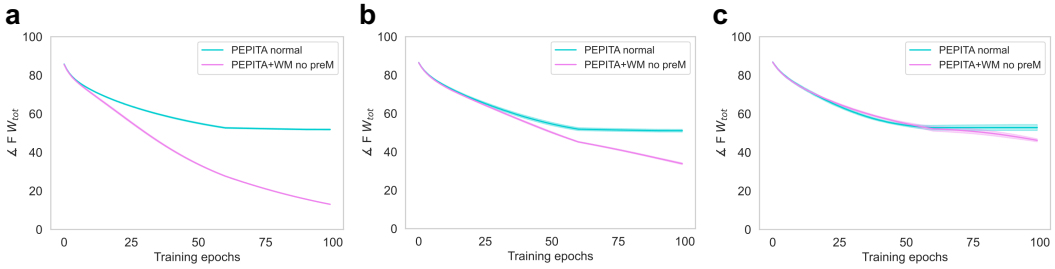

Figure S8: Alignment angle between F and W_{tot} during training with (pink curve) or without (blue curve) WM for the CIFAR-10 dataset for (a) 1-, (b) 2-, (c) 3-hidden-layer networks. The hyperparameters are reported in Table S3 for the 1-hidden-layer network and Table S4 for the 2- and 3-hidden-layer networks. The plots indicate mean and standard deviation over 10 independent runs.