Exploring the potential of Direct Feedback Alignment for Continual Learning

Anonymous authors
Paper under double-blind review

Abstract

Real-world applications of machine learning require robustness to shifts in the data distribution over time. A critical limitation of standard artificial neural networks trained with backpropagation (BP) is their susceptibility to catastrophic forgetting: they "forget" prior knowledge when trained on a new task, while biological neural networks tend to be more robust to catastrophic forgetting. While various algorithmic ways of mitigating catastrophic forgetting have been proposed, developing an algorithm that is capable of learning continuously remains an open problem. Motivated by recent theoretical results, here we explore whether a biologically inspired learning algorithm like Direct Feedback Alignment (DFA) can mitigate catastrophic forgetting in artificial neural networks. We train fully-connected networks on several continual learning benchmarks using DFA and compare its performance to vanilla back propagation, random features, and other continual learning algorithms. We find that an inherent bias of DFA, called "degeneracy breaking", leads to low average forgetting on common continual learning benchmarks when using DFA in the Domain-Incremental learning scenario and in the Task-Incremental learning scenario. We show how to control the trade-off between learning and forgetting with DFA, and relate different modes of using DFA to other methods in the field.

1 Introduction

Neural networks have demonstrated remarkable achievements in various domains, including image recognition (Krizhevsky et al., 2012; LeCun et al., 2015; Simonyan & Zisserman, 2015; He et al., 2016; Dosovitskiy et al., 2021) and natural language processing (Devlin et al., 2019; Howard & Ruder, 2018; Radford et al., 2018; Brown et al., 2020; OpenAI, 2024). However, a significant drawback of traditional neural networks is their susceptibility to catastrophic forgetting (Goodfellow et al., 2015; McCloskey & Cohen, 1989). Catastrophic forgetting refers to a loss in performance on previously learned tasks when the network is trained on new tasks or data distributions. This limitation impedes the ability of neural networks to continuously learn and adapt to evolving environments, limiting their practical applicability in real-world scenarios.

Continual learning (CL) focuses on developing algorithms and techniques that enable systems to maintain old knowledge, while also being able to learn new information. This requires resolving the "stability–plasticity dilemma" (Carpenter, 1986; Mermillod et al., 2013): a model needs plasticity to obtain new knowledge and adapt to new environments, while also requiring stability to prevent forgetting of previous information. Typical approaches to CL include dynamic architectures (Razavian et al., 2014), progressive learning (Fayek et al., 2020), regularisation techniques (Kirkpatrick et al., 2017; Zenke et al., 2017), or episodic memory replay (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Rebuffi et al., 2017; Shin et al., 2017; Kamra et al., 2017; Seff et al., 2017); see (De Lange et al., 2021) for a review.

Meanwhile, biological neural networks do not suffer from catastrophic forgetting nearly as badly as artificial neural networks. Consequently, several approaches to mitigate catastrophic forgetting have taken direct inspiration from biology: for example, the complexity of synaptic plasticity in biological neurons inspired regularisation approaches such as Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) and Synaptic Intelligence (Zenke et al., 2017).

Here, we continue this line of thought by investigating the potential of a biologically plausible learning algorithm to mitigate catastrophic forgetting: Direct Feedback Alignment (DFA). Proposed by Nøkland (2016) as a biologically plausible algorithm to train neural networks, DFA propagates the error signal through fixed, random feedback connections directly to the hidden layers, thereby solving the weight transport problem that plagues vanilla backpropagation (Grossberg, 1987; Crick, 1989). Decoupling the weight updates of different layers enables parallel and local updates, which are considered key features of synaptic updates in the brain (Bengio et al., 2015).

We focus on the potential of DFA not just because of its greater biological plausibility compared to vanilla backpropagation. We are encouraged by the recent analysis of Refinetti et al. (2021), who showed that DFA has a *degeneracy breaking* property. To learn with DFA, the neural network has to first align its weights (to some extent) with the feedback matrices to ensure that the error signal can be backpropagated efficiently (Lillicrap et al., 2016). This alignment has the effect that neural networks trained by DFA always converge to the same region in the loss landscape, independently of their initialisation, and in contrast to networks trained by vanilla back propagation. In other words, DFA will drive a neural network to a specific region in the loss landscape. The location of this region depends on the feedback matrices, thereby breaking the degeneracy of the solutions of vanilla SGD. In this work, we ask whether we can use the influence of the DFA feedback matrix on the weights learnt by a neural network to mitigate catastrophic forgetting. We formulate and test two different hypotheses for how DFA can facilitate continual learning.

Our **first hypothesis** is that using the *same* feedback matrix for different tasks in a continual learning curriculum prevents catastrophic forgetting by implicitly biasing the weights to a single region of loss landscape, as they always need to align with the same feedback matrix. In this case, DFA would act as a an *implicit regulariser*, similar to other algorithms like Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017), a popular implicit regularisation technique. We will call this approach **DFA-same**.

Our **second hypothesis** is that using *different* feedback matrices for each task will effectively update the weight matrices in different directions for each task, thus preventing catastrophic forgetting. This **DFA-diff** approach is inspired by various CL algorithms that explicitly orthogonalise gradients (Zeng et al., 2019; He & Jaeger, 2018; Bennani & Sugiyama, 2020). This idea can also be motivated from a neuroscientific perspective given recent evidence that neural population codes orthogonalize with learning (Flesch et al., 2022; Failor et al., 2021; Zeng et al., 2019).

Since efficient training of convolutional neural networks (CNNs) with DFA remains an open problem (Crafton et al., 2019; Launay et al., 2020), we focus on fully-connected networks and limit ourselves to simple benchmark datasets based on MNIST and FMNIST where these architectures achieve good performance. In particular, we use the split and the permuted manipulations of MNIST and FMNIST that turn the 10-class classification tasks into a set of successive classification tasks (Kirkpatrick et al., 2017; Zenke et al., 2017).

The contributions of this paper are threefold:

1. We empirically show that DFA performs better at Continual Learning than vanilla back-propagation and other baselines, such as random features (RF) and Elastic Weight Consolidation (EWC).

2. We benchmark the performance of different DFA strategies for Continual Learning, either sharing or changing feedback matrices between tasks, and show how the best strategy depends on the type of continual learning task and the network architecture.

3. We show that the scale of the feedback matrix allows one to trade-off plasticity vs. stability in continual learning, beyond what can be achieved by choosing layer-specific learning rates.

Exploring the potential of DFA to mitigate catastrophic forgetting both deepens our understanding of the working mechanism behind DFA by testing it in a continual learning setting, and it adds to the tool set that can be tested on continual learning problems. The remainder of this paper is organized as follows: In section 2, we provide the details of the experimental setup. Section 3 presents our results by benchmarking DFA on various continual learning tasks and contrasting its performance with various baselines. Section 4 is dedicated to a discussion of our results and gives some concluding perspectives.

2 Methods

Direct Feedback Alignment (DFA) DFA presents a departure from the traditional backpropagation algorithm (Rumelhart et al., 1988), which relies on weight symmetry and precise weight updates for accurate error propagation. Instead, DFA employs random feedback matrices that are decoupled from the forward weight matrices, enabling more flexible weight updates without the constraints imposed by weight symmetry (Nøkland, 2016).

To state the DFA weight updates clearly, and to contrast them with vanilla backpropagation, we consider mini-batches of input-output pairs (x, y) that we want the network to learn. For simplicity, we consider a simple network composed of two fully-connected layers and one softmax layer at the end. We denote as W_i, h_i and a_i the weight matrix, activation function and activations of layer i; The output of the network is $\hat{y}$ and the activations of the output layer are a_y. The error e is the derivative of the loss J and in case of a cross-entropy loss it is equal to:

$$e = \delta a_y = \frac{\partial J}{\partial a_y} = \hat{y} - y \tag{1}$$

Denoting $\odot$ as the Hadamard product, for BP (on the left) and DFA (on the right), the gradients of the hidden layers are calculated as

BP	DFA	
$\delta W_3 = -eh_2^T$	$\delta W_3 = -eh_2^T$	(2)
$\delta W_2 = \frac{\partial J}{\partial a_2} = (W_3^T e) \odot f'(a_2) h_1^T$	$\delta W_2 = \frac{\partial J}{\partial a_1} = (B_2 e) \odot f'(a_2) h_1^T$	(3)
$\delta W_1 = (W_2^T \delta a_2) \odot f'(a_1) x^T$	$\delta W_1 = (B_1 e) \odot f'(a_1) x^T$	(4)

While the final layer (the readout layer) is updated in the same way with both algorithms, the weight updates for the other layers are all different. BP implements the exact gradient for each layer by applying the chain rule to compute the derivatives of the loss; instead, DFA keeps only the error term and substitutes the derivative of the following layer by an entry of the feedback matrix. A detailed theoretical analysis of the evolution of the update dynamics can be found in Refinetti et al. (2021).

In the case of DFA-same, the feedback matrices (B_1 and B_2) are initialized according to a uniform distribution and kept unchanged during the training of all tasks. According to the degeneracy breaking of DFA discussed in the introduction, DFA-same could alleviate catastrophic forgetting by keeping the weights of the network when training on the second task close to the weights learnt on the first task, etc.

In DFA-diff, we use a different feedback matrix for each task, by sampling the new ones from the same Uniform distribution used to sampled the first one. Our hypothesis is that the different feedback matrices may bias the dynamics of the networks in orthogonal directions of the weight landscape for every task. In fact, due to the large dimensions of the matrix and their sampling from uniform distributions, the new feedback matrices are approximately orthogonal to the previous ones. The gradient alignment will then be directed to orthogonal directions for every task. Making the gradients explicitly orthogonal to each other is known to help against catastrophic forgetting, and the idea is exploited in methods such as Orthogonal Weight Modification (OWM) (Zeng et al., 2019), conceptor-aided backprop (He & Jaeger, 2018), and orthogonal gradient descent (Bennani & Sugiyama, 2020), because the features of different tasks are learned along orthogonal manifolds, and the weights updates do not interfere with the previous ones. Appendix A confirms that the gradients are indeed orthogonal when two networks are trained on the same dataset with orthogonal feedback matrices.

Benchmark datasets We report results on the FashionMNIST (FMNIST) dataset (Xiao et al., 2017), but our findings are consistent with the MNIST dataset, see appendix C. We do not apply data augmentation techniques, and train the networks on two different CL benchmark tasks: In our experimental evaluations, we use:

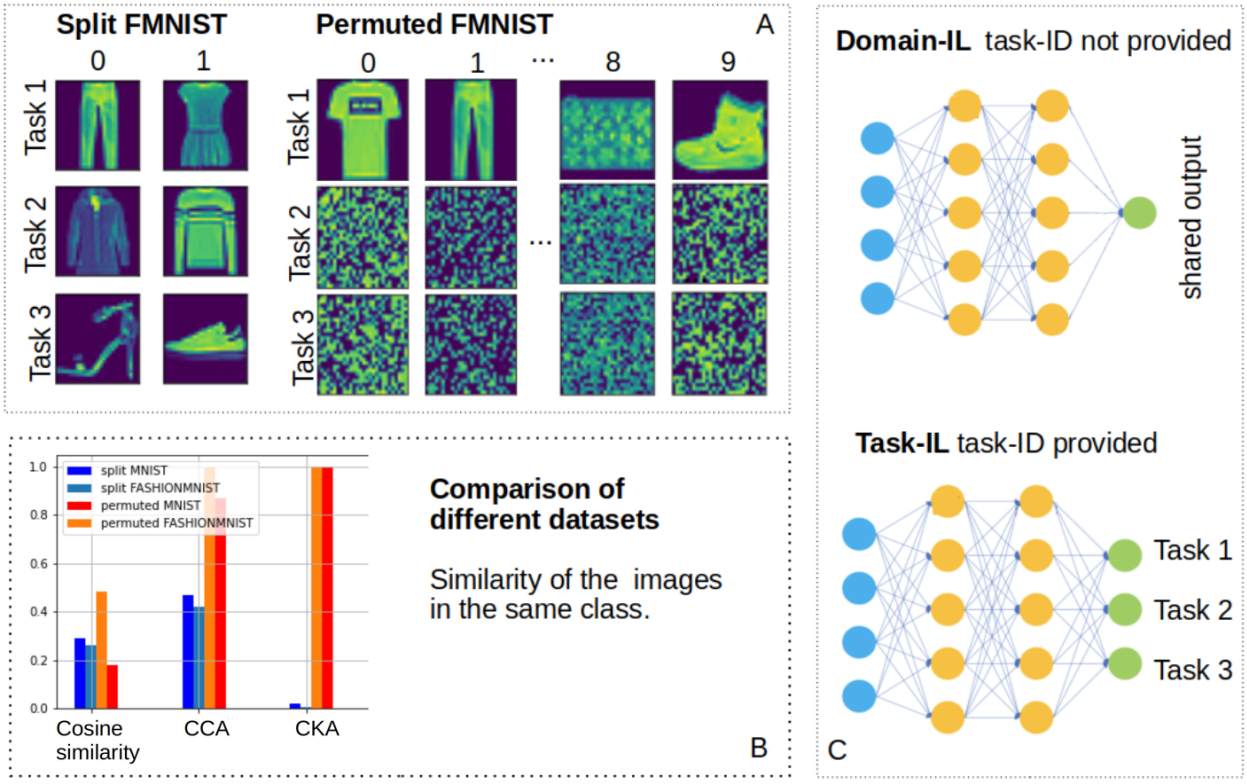

Figure 1: **A** One example of the first three tasks in the split (Binary classification) and permuted dataset (classification). **B** Similarity measures of the images in the same class averaged over the different classes of the Split and Permuted datasets. The permuted-FMNIST dataset is the one where the images have the highest similarity among tasks, while split-FMNIST has very different images in the same class. See section appendix E for a description of the similarity measures. **C** One example of a three-layer architecture used in the Domain-IL and Task-IL scenarios. In the second case, one output node is dedicated to training and evaluating one specific task.

1. Permuted FMNIST (pFMNIST), where we generate a sequence of learning tasks by permuting the pixels of each image; the idea here is to test a model's ability to incrementally learn new information with similar feature representations (Kemker et al., 2017).

2. Split FMNIST (sFMNIST), where we split the original dataset of 10 classes into five smaller datasets with two disjoint classes for each. The resulting smaller datasets will have different modalities, so a model trained sequentially on them needs to be able to incrementally learn new information with dramatically different feature representations.

We show an example of the split and permuted FMNIST in fig. 1, panel A.

Experimental setup We test DFA in two continual learning scenarios (van de Ven & Tolias, 2019) (Panel C, fig. 1):

1. In the Domain-IL scenario, the network does not have the information about the task-ID during inference. The whole architecture is shared among the tasks, including the output layer.

2. In the Task-IL scenario, the network can use the task-ID during both training and testing. We implement this approach by having task-specific output layers that are updated only during the training on the corresponding task and used whenever testing on the same task.

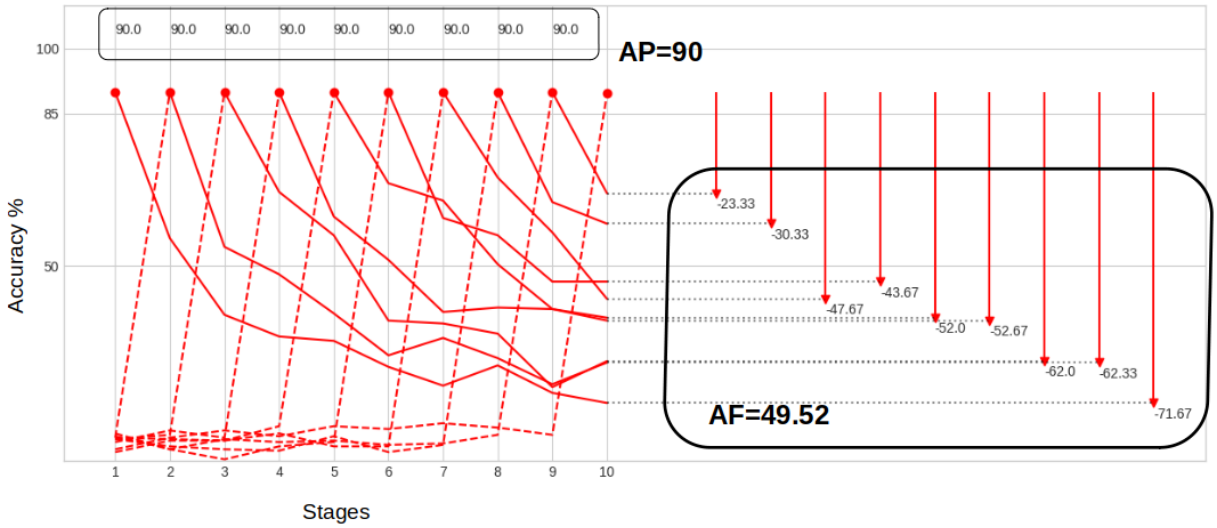

Figure 2: Illustration on Average Forgetting (AF) and Average Performance (AP) in the case of backpropagation on the p-FMNIST task in the Task-IL scenario, where the network has a separate head for each task. AP describes the test accuracy of the network, averaged over the tasks, while AF quantifies the average of the difference in accuracy on a given task after training on it and at the end of the whole training trajectory.

We show in Figure 4 and in appendix A that the weight alignment and gradient alignment of DFA are preserved if the output layer is re-initialized. These tests are important because the degeneracy breaking feature of DFA was previously observed in networks with weights sampled from independent distributions, while in this case only the output layers of the networks in comparison are different.

In our experiments, we use 3-layer Fully-Connected Networks with 1000 neurons in each hidden layer. We train the networks for a maximum of 1000 epochs and apply early-stopping by halting the training as soon as the network overcomes 99% training accuracy. All layers are initialized using the Xavier uniform initialization (Glorot & Bengio, 2010). We choose a logistic activation function in the output layer and ReLU in the other layers. The loss function is cross-entropy.

Evaluation metrics Performance in continual learning is difficult to report by one single measure due to the stability–plasticity trade-off. We will summarize the performances of the algorithms under study using average performance (AP) and average forgetting (AF) (Lopez-Paz & Ranzato, 2022; Chaudhry et al., 2018), which we visualize in fig. 2. To define AF and AP, we consider a number of tasks T and denote by Acc_{ij} the accuracy of the network on the jth task at the end of training on the ith task, where $j, i \in \{1, \dots, T\}$. AP quantifies the performance of the network by calculating the mean test accuracy of the network on each task, directly after training on it:

$$\text{AP} = \frac{1}{T} \sum_{i=1}^{T} \text{Acc}_{i,i}. \tag{5}$$

AF measures how much the model loses accuracy on previous tasks during continual learning:

$$\text{AF} = \frac{1}{T-1} \sum_{i=1}^{T-1} \text{Acc}_{i,i} - \text{Acc}_{i,T} \tag{6}$$

Baselines and hyper-parameters tuning To benchmark the performance of DFA, we consider the following additional baselines:

1. Backpropagation is the most important baseline: we train the same fully-connected networks, adding a Dropout layer (Srivastava et al., 2014) after each layer, which we find helps with generalization,

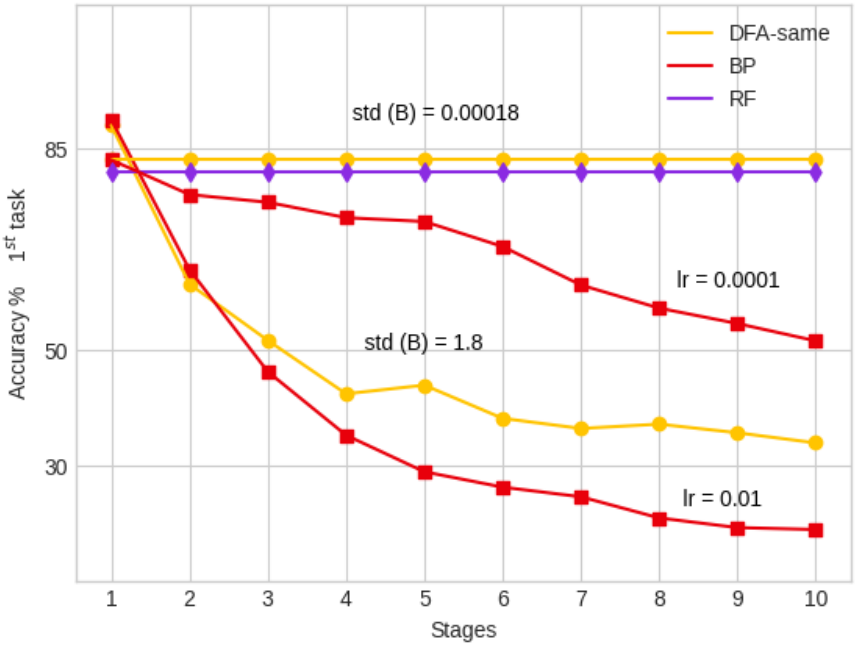

Figure 3: Performance of the first task throughout learning all tasks evaluated on the permuted-FMNIST (above) in the Task-IL scenario. DFA-same is more powerful than RF (purple line). Compared to BP, DFA-same has less forgetting in both cases: when we optimize the models for maximum performance and for least forgetting.

in accordance to the literature (Mirzadeh et al., 2020). We perform a grid-search optimization for the learning rate in the range between 1e-2 and 1e-4.

2. Random Features (Rahimi & Recht, 2008; 2009) are fully-connected networks in which we train only the readout layer with BP. The first and the intermediate layers are kept unchanged during the training phase. This model can be applied only in the Task-IL scenario because it requires a different output layer for each task in the testing phase. The motivation for using this model is to understand the performance of a neural network that does *not* learn data-dependent features, akin to the lazy regime (Chizat et al., 2019). We use a learning rate of 1e-2.

3. Elastic weight consolidation (EWC) (Kirkpatrick et al., 2017) adds a regularisation term on the loss that penalizes the change of the weights that are more important for previous tasks. It achieves this by pre-multiplying the BP weight updates using the inverse of the diagonal approximation of the Fisher information matrix of the model. In our EWC experiments, we chose a learning rate of 1e-3 and an "importance" of 1000, which is another important hyper-parameter for EWC.

The error bars in table 1 are computed over a repetition of the experiment with 5 random seeds.

3 Results

3.1 DFA between the two ends of the plasticity-stability trade-off

Catastrophic forgetting arises in Backpropagation (Rumelhart et al., 1988) due to the fact that the weights of the network are updated in the direction of the minimum of the loss, irrespective of the previous tasks. In this

Table 1: Results in terms of Average Performance (AP) and Average Forgetting (AF) in the different datasets and scenarios (Task-IL and Domain-IL) for all the methods (DFA–diff, DFA–same, BP, BP–ablated, EWC, RF). In the first column, the networks are optimized for maximum performances and in the second column for minimum forgetting. We discuss all these results in detail in section 3 and we present them visually in Figure 5. In the case of minimized forgetting, we report here the minimum AP % values above the RF baseline.

		AP	AF	AP	AF
		Optimized for performance		Minimizing forgetting	
Split FMNIST Task-IL	DFA-diff	100 ± 0	9.8 ± 2	93.9 ± 0.1	0 ± 0
	DFA-same	100 ± 0	29.4 ± 1.2	95.4 ± 0	0 ± 0
	BP	100 ± 0	15 ± 2.3	89.5 ± 0.1	0.3 ± 0.3
	BP ablated	100 ± 0	9.7 ± 1.6	94 ± 0	0 ± 0
	EWC	99.2 ± 0.2	4.5 ± 1.5	99.1 ± 0.2	3.4 ± 1.8
	RF	93.4 ± 0.1	0 ± 0	93.4 ± 0.1	0 ± 0
Permuted FMNIST Task-IL	DFA-diff	88.5 ± 0.1	44.5 ± 2.4	83 ± 0	0 ± 0
	DFA-same	88.2 ± 0.1	50.1 ± 2.5	83 ± 0	0 ± 0
	BP	90.0 ± 0.1	49.5 ± 2.5	83 ± 0	11.8 ± 0.8
	BP ablated	90 ± 0.1	47 ± 2	82.8 ± 0.1	0 ± 0
	EWC	88.7 ± 0.1	46.5 ± 2.7	83 ± 0	8.7 ± 1.2
	RF	80.9 ± 0.1	0 ± 0	80.9 ± 0.1	0 ± 0
Split FMNIST Domain-IL	DFA-diff	100 ± 0	45.8 ±2.4	94.0 ± 0	35.56 ± 1.6
	DFA-same	100 ± 0	35.5 ± 0.2	98.2 ± 0	32.1 ± 1.6
	BP	100 ± 0	37.8 ± 0.4	94.5 ± 0.1	35.8 ± 0.6
	BP ablated	100 ± 0	40.5 ± 1.7	98.4 ± 0.1	34.7 ± 0.7
	EWC	100 ± 0	46.9 ± 4.8	98.7 ± 0.4	40.4 ± 3.1
Permuted FMNIST Domain-IL	DFA-diff	88.3 ± 0	71.8 ± 1.2	82.3 ± 0.1	29.1 ± 0.9
	DFA-same	88.2 ± 0.1	45.8 ± 1.9	85.8 ± 0	26.2 ± 1.9
	BP	89.7 ± 0.1	57.6 ± 1.1	85.4 ± 0.1	23.1± 1.5
	BP ablated	88.7 ± 0.1	49.4 ± 0.4	84 ± 0.1	26.2 ± 2.4
	EWC	80.7 ±0.2	55.7 ± 2.5	80.7 ± 0.2	55.7 ± 2.5

case the network is flexible to fit the task at hand, but forgetting is high. Random features are at the other extreme, since only the weights in the output layer are trained on the task it results in a more rigid method that retains previous performances. BP and random features therefore delineate the two extremes between which we would like to interpolate: we would like DFA to be less susceptible to catastrophic forgetting than BP, while being able to learn data-dependent features that allow it to beat the performance of random features. We report the results of our experiments that we obtain by optimizing the hyper-parameters for maximum AP or for minimum AF in table 1.

When we train DFA and BP at the same training conditions, we can see that DFA can reach the same performance of 100% accuracy as BP on the split dataset (where the tasks are binary classifications) and almost reach it on the permuted dataset (where the tasks consist of 10-class classifications), where DFA achieves 88.2 ± 0.1 [1] and BP 89.7 ± 0.1. Thus, DFA proves to be able to adapt the network to learn the new tasks efficiently. In terms of forgetting, DFA achieves 10% less AF than BP in both split and permuted

[1]We notice that letting DFA train for more epochs, it can achieve the same level as BP, but we decided to keep a maximum of 1000 epochs for all methods.

datasets in the Domain-IL scenario. This trend is conserved when the experiments are performed with the MNIST datasets (see appendix C). The advantage of DFA in the Domain-IL scenario is in accordance to the hypothesis for which DFA learns close representations for different tasks. In fact, the Domain-IL has a unique output layer shared among all the tasks and this requires similar representations among different tasks in order to correctly classify the previous ones.

In the Task-IL scenario, the average forgetting of BP improves with respect to the Domain-IL scenario by 8% and 22% for the Split and Permuted FMNIST datasets respectively. On the other hand, DFA improves only by 6% on the split dataset and become worse, with AF from 45.8% to 50.1%, in the permuted-FMNIST. This trend brings DFA to perform worse than BP on the split dataset and comparably to BP on the permuted dataset[2]. The accuracy trend of DFA can be explained in light of our hypothesis: if the output layer is specialized for every task, keeping the representations similar to each other can bring a disadvantage, as the previous representations are "overwritten".

Regarding the comparison of Random Features and DFA with minimized forgetting, we find that DFA can achieve a better Average Performance than a Random Feature model, both in the split and permuted datasets, while maintaining zero forgetting, just as RF. Specifically, DFA shows AP at least 2% higher than RF (see table 1, column "Minimizing forgetting"). One example is shown in fig. 3, where DFA reaches 83% accuracy while Random Features reaches 80.9%. The fact that DFA reaches higher performances than RF means that the weights of the first two layers are updated, and the null forgetting values DFA means that this weights update can happen in such a way that the information of the previous tasks is conserved.

Overall, we find that DFA can embody different behaviours according to the value of its hyper-parameters. It can almost reach the same generalization performances as BP while displaying less forgetting in the majority of the cases; and it can achieve zero forgetting like RF, while generalizing *better* than RF. It thus occupies a sweet-spot between the plastic BP and the static RF.

3.2 Extension of DFA to the Task-IL scenario

We showed that the standard DFA cannot benefit from the introduction of a specialized output layer for every task (Task-IL scenario). We propose an extension of DFA in which there is a different feedback matrix for every task: in fig. 4, we show that in this setting the representations of one dataset are learned in different manifolds. According to our hypothesis, this phenomenon can be extended to the learning of different datasets and this would allow to learn new tasks with less interference with previous representations. We expect that the combination of a dedicated feedback matrix alongside a dedicated output layer can reduce forgetting in the Task-IL scenario. In the remaining, we will denote the standard DFA as DFA-same and this extension as DFA-diff.

We find that the main differences with respect to DFA-same are in the Average Forgetting measure and are in accordance to the hypothesis: when optimized for performance and in the Task-IL scenario, DFA-diff has $9.8\% \pm 2\%$ AF. This value is around 20% lower than the one of DFA-same and 5% lower than BP's. In the Domain-IL scenario, on the other hand, DFA-diff has at least 8% more forgetting than BP and 10% more than DFA-same. When optimized for minimal forgetting, DFA-diff also achieves null forgetting in the Task-IL scenario, but it shows lower performances than DFA-same (up to 4.2% less than DFA-same in the split Domain-IL).

3.3 Comparing DFA with Elastic weight consolidation

In the Domain-IL scenario there is an advantage of DFA-same over EWC of about 10% AF. This could be due to the underlying hypothesis that in DFA-same the implicit regularisation given by the feedback matrix is kept constant. On the other hand, the regularisation factor in EWC is different for every new task, as explained in section 2. On the permuted dataset, EWC has a performance that is intermediate between

[2]The comparable value of AF between DFA and BP in the Task-IL permuted case hinders the advantage of DFA for the forgetting of the earliest tasks. We show in fig. 3 the trend of forgetting of the first task and we can notice a clear advantage. In **??** we report the forgetting trends of all the tasks, where one can see that DFA forgets more tasks seen one or two epochs before, but is more stable for earlier ones.

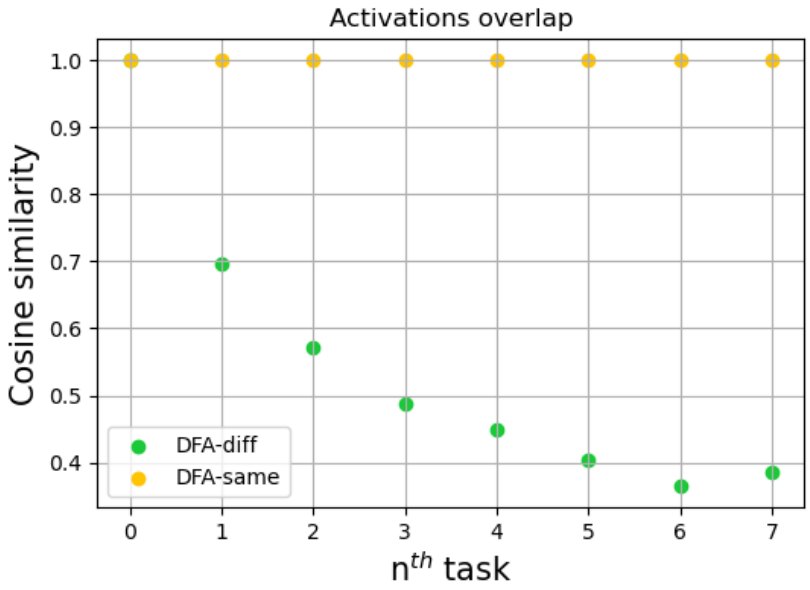

Figure 4: Cosine similarity of the change in activations due to the n^{th} training vs the change since the first initialization. Values Averaged over the first two layers of the network and the images of the test dataset. The network is trained repeatedly on the same dataset (FMNIST) while the output layer is re-initialized at the beginning of every session. The overall change in activation in DFA-same is always aligned with the change taking place in every single training, while in DFA-diff it is less and less aligned with the changes due to subsequent trainings.

DFA-same and DFA-diff. In fact, EWC can be considered a method in between DFA-same and DFA-diff because the "pulling forces" in the loss due to the feedback matrices will not be parallel to the ones of the previous tasks as in DFA-same, but not orthogonal as in DFA-diff, either. On the split datasets, on the other hand, EWC tends to be better than DFA-diff. In the Task-IL scenario EWC is the best method both when we evaluate on MNIST (see appendix C) and FMNIST. In the Domain-IL it is the best only in the experiments on MNIST. This might be due to a difficulty to find the correct hyper-parameters that would render EWC performative on FMNIST.

3.4 Ablation experiments

In order to investigate the reasons behind the success of DFA in some of the CL scenarios, we look for the underlying mechanism that influences DFA's stability to forgetting the most. For achieving minimum AF, we performed a grid-search over the hyper-parameters and we find that forgetting is reduced with smaller variance of the feedback matrix for both DFA-same and DFA-diff, see fig. 5. This impact can be easily explained by observing that the entries of the Feedback matrix are a multiplicative factor in the update rule of the layers 1 and 2 (see section 2, in the right column of equations 3 and 4); thus the smaller the values in the matrix, the smaller will be the update of the weights of these two layers. The output layer is updated exactly as in BP, so this layer is not affected by the variance of the Feedback matrix. In the limit of a matrix filled with zeroes, DFA approaches a Random Feature behaviour where only the readout layer is updated. We therefore performed an ablation experiment with BP where we reduced the learning rate of BP in the first two layers to mimic the effect of feedback matrices with reduced variance, while keeping the learning rate of the readout layer fixed and similar to the one used for DFA. We show the results of the ablated experiment in all the datasets in fig. 5. We found that lowering the learning rate of the intermediate layer indeed brings BP towards lower AF and lower AP, similarly to the effect of lowering the variance in DFA's feedback matrix. In fact, BP with reduced learning rates can surpass DFA's stability in some cases, for example in the Task-IL, permuted case. In all the other cases, BP-ablated can reach AF similar to DFA,

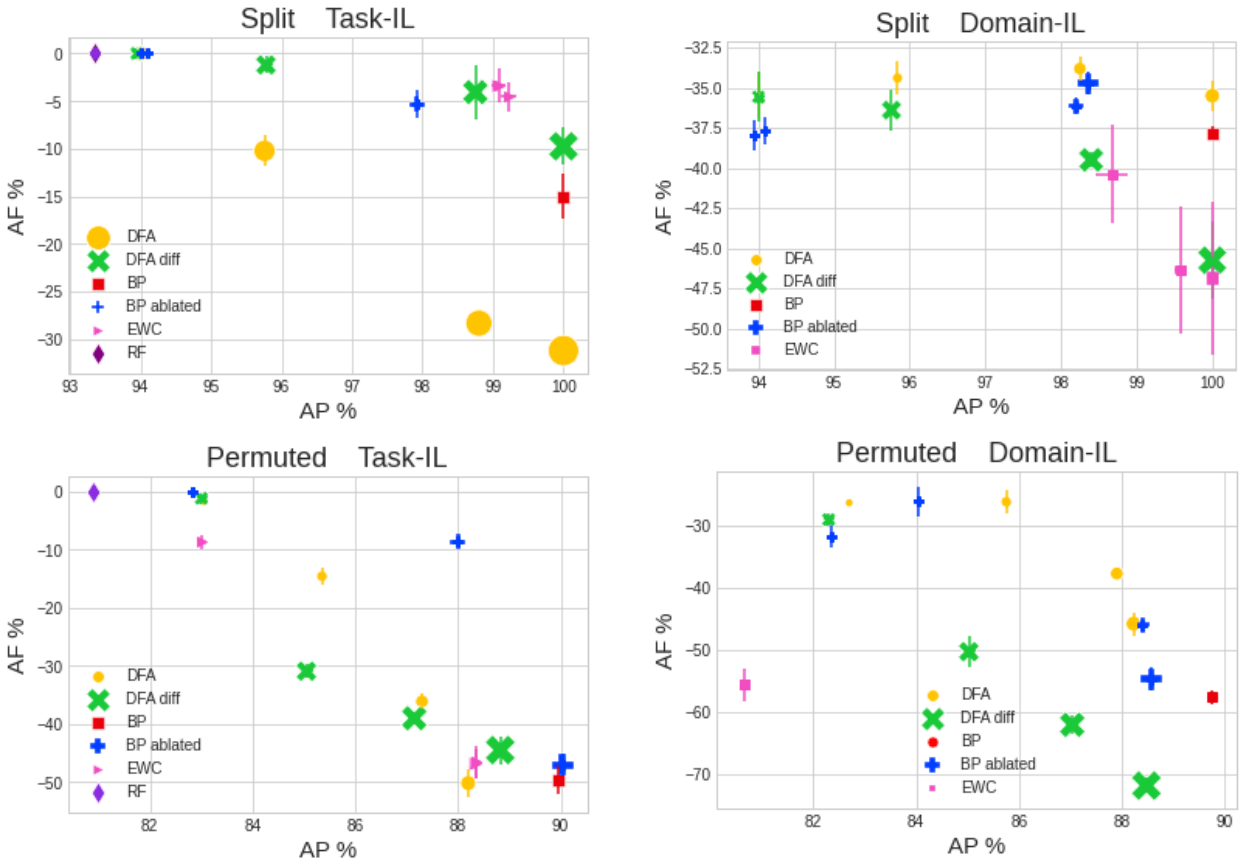

Figure 5: AF-AP plots for all datasets (by rows) and scenarios (by columns). In each plot, the increasing size of the marker in DFA and DFA-diff indicates an increase in the variance of the feedback matrix (from 1.8×10^{-3} to 1.8) and for BP an increase of the learning rate of the first two layers (from 10^{-8} to 10^{-3}); the learning rate of the third layer is kept to 0.01 to match the value used in DFA. We can notice that for larger values of the feedback matrix variance, DFA can reach higher AP at the price of smaller AF (smaller dots moving towards the top-left part of the plots, which culminates in the RF regime). The same trend is followed by BP when the learning rate of the first two layers decreases (BP ablated).

underlining the importance of variance of the feedback matrix for the stability of DFA. Crucially, DFA still retains higher accuracy for old tasks in the long run: As we show in appendix B, figure fig. 7, the accuracy drop for the case of Task-IL p-FMNIST in DFA is bounded to a maximum of 59% while BP-ablated reaches 67% forgetting.

4 Conclusions

We have explored the potential of a biologically inspired learning algorithm, direct feedback alignment, to alleviate catastrophic forgetting in artificial neural networks. By comparing the performance of DFA and various baseline algorithms on several benchmarks, we found that DFA has potential as an algorithm for continual learning. For example, we found that DFA-same can alleviate catastrophic forgetting better than BP and EWC in the Domain-IL scenario. This result is consistent with the first hypothesis, whereby DFA-same imposes weight alignment in the presence of different datasets. We saw that the networks are indeed able to exploit this for mitigating catastrophic forgetting. According to our hypothesis, DFA-same works as an implicitly regularised method like EWC, but with the advantage of keeping exactly the same constraints on the weights, while EWC adjusts the constraint of the weights after every new dataset encountered.

Empirical evaluation also shows that DFA with different feedback matrices for each task in the curriculum has an advantage with respect to EWC and BP in the Task-IL scenario. This is the architectural scenario in which the output layer is specific for every task. This allows learning the features on different manifolds in weight space, while still permitting an efficient encoding of the features for a successful classification of the different tasks. In continual learning, when the features are learned on different manifolds, the learning of the new tasks does not interfere with the separating hyperplane of the previous tasks, so it can be an advantage in the Task-IL scenario. The drastic improvements of DFA-diff in the Task-IL scenario in contrast to the minor improvement of DFA-same in this setting corroborates our second hypothesis, whereby gradient alignment extends to the case of different datasets, allowing DFA-diff to learn in different manifolds. This approach can thus be employed for an effective mitigation to catastrophic forgetting. Finally, we tested gradient alignment in two specific experiments, and we find it visible in the case in which DFA is trained on the same series of datasets starting from different initialisations with the same feedback matrices (appendix A, panel C) and when DFA is trained on the same dataset repeatedly but with different Feedback Matrices (Figure 4).

While backpropagation with layer-specific learning rates in the first two layers (BP ablated) has a significantly lower overall forgetting than DFA-same in the case of Task-IL and permuted dataset, DFA-same retained its advantage with respect to BP and BP ablated for the tasks that were processed at the beginning (see **??** and fig. 3). The design of BP ablated itself was inspired by the impact of the scale of DFA's feedback matrix on forgetting. The advantage of DFA-same over this method is further evidence that weight alignment is taking place and is beneficial against catastrophic forgetting beyond the impact of the Feedback matrix on the learning rate.

In the future, it will be interesting to investigate how the architecture of the networks, and specifically the width and the depth of the networks influence the plasticity-stability trade off curves. In the case of the split dataset, the forgetting is heavily dependent on the similarity of consecutive tasks (see larger error bars in fig. 5). On top of curriculum learning, it might also be beneficial to use convolutional layers before the Fully–Connected part to replace the highly-variable distribution of pixels with more rationalized features (Crafton et al., 2019). We tested the Continual Learning scenarios where the model is given the task–ID at test time (Task-IL) or not (Domain-IL). The next challenge is the class–incremental learning scenario where the model has to provide the task-ID when solving a task as is the case, for example, in many reinforcement learning scenarios. Increasing the biological plausibility of DFA is another interesting direction. Investigating the viability of DFA extensions that go toward more local update rules like the one analysed by Dellaferrera et al. (2021) would be particularly interesting.

References

Yoshua Bengio, Dong-Hyun Lee, Jörg Bornschein, and Zhouhan Lin. Towards biologically plausible deep learning. *ArXiv*, abs/1502.04156, 2015.

Mehdi Abbana Bennani and Masashi Sugiyama. Generalisation guarantees for continual learning with orthogonal gradient descent. *ArXiv*, abs/2006.11942, 2020.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Gail Carpenter. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, 11 1986. doi: 10.1016/S0734-189X(87)80014-2.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. *Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence*, pp. 556572. Springer International Publishing, 2018. ISBN 9783030012526. doi: 10.1007/978-3-030-01252-6_33.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *ArXiv*, abs/1812.00420, 2019.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 2933–2943. Curran Associates, Inc., 2019.

Brian Crafton, Abhinav Parihar, Evan Gebhardt, and Arijit Raychowdhury. Direct feedback alignment with sparse connections for local learning, 2019.

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

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Giorgia Dellaferrera, Stanislaw Wozniak, Giacomo Indiveri, Angeliki Pantazi, and Evangelos Eleftheriou. Learning in deep neural networks using a biologically inspired optimizer, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Samuel W. Failor, Matteo Carandini, and Kenneth D. Harris. Learning orthogonalizes visual cortical population codes. *bioRxiv*, 2021. doi: 10.1101/2021.05.23.445338.

Haytham M. Fayek, Lawrence Cavedon, and Hong Ren Wu. Progressive learning: A deep learning framework for continual learning. *Neural networks : the official journal of the International Neural Network Society*, 128:345–357, 2020.

Timo Flesch, Keno Juechems, Tsvetomira Dumbalska, Andrew Saxe, and Christopher Summerfield. Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron*, 110(7):1258–1270.e11, 2022. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2022.01.005.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks, 2015.

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

David Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16:2639–64, 01 2005. doi: 10.1162/0899766042321814.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation. In *International Conference on Learning Representations*, 2018.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.

Nitin Kamra, Umang Gupta, and Yan Liu. Deep generative dual memory network for continual learning. *ArXiv*, abs/1710.10368, 2017.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks, 2017.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2017.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019.

A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Julien Launay, Iacopo Poli, François Boniface, and Florent Krzakala. Direct feedback alignment scales to modern deep learning tasks and architectures. *Advances in neural information processing systems*, 33: 9346–9360, 2020.

Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015. doi: 10.1038/nature14539.

Timothy P. Lillicrap, Daniel Cownden, Douglas Blair Tweed, and Colin J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7, 2016.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NIPS*, 2017.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning, 2022.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pp. 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8.

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4:504, 08 2013. doi: 10.3389/fpsyg.2013.00504.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning, 2020.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *NIPS*, 2016.

OpenAI. Gpt-4 technical report, 2024.

Alec Radford, Karthik Narasimhan, et al. Improving language understanding by generative pre-training, 2018.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pp. 1313–1320, 2009.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 512–519, 2014.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2017.

Maria Refinetti, Stéphane d'Ascoli, Ruben Ohana, and Sebastian Goldt. Align, then memorise: the dynamics of learning with feedback alignment. *Journal of Physics A: Mathematical and Theoretical*, 55, 2021.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning representations by back-propagating errors*, pp. 696699. MIT Press, Cambridge, MA, USA, 1988. ISBN 0262010976.

Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual learning in generative adversarial nets. *ArXiv*, abs/1705.08395, 2017.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NIPS*, 2017.

K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

Gido M. van de Ven and Andreas S. Tolias. Three scenarios for continual learning, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364372, August 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0080-x.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 70:3987–3995, 2017.

A Degeneracy breaking of DFA

Previous work on DFA (Refinetti et al., 2021) and FA (Lillicrap et al., 2016) has shown that DFA brings the gradients and the weights to align with the feedback matrix. This phenomenon is called degeneracy breaking because among all the combinations of the weights that can fit the dataset, DFA selects one that is closest to the one aligned to the feedback matrix. **Gradient Alignment** and **Weight Alignment** were previously measured on the same dataset, starting from different initialization seeds. Here, we show the overlap of the weights (in panel A) and the overlap of the gradients (in panel B) among networks with the same/different initialization seed on the weight (yellow/purple in the first column of panel A) and the same/different feedback matrix (DFA-same/DFA-diff). In the training on Task 1 all the networks have the same feedback matrix.

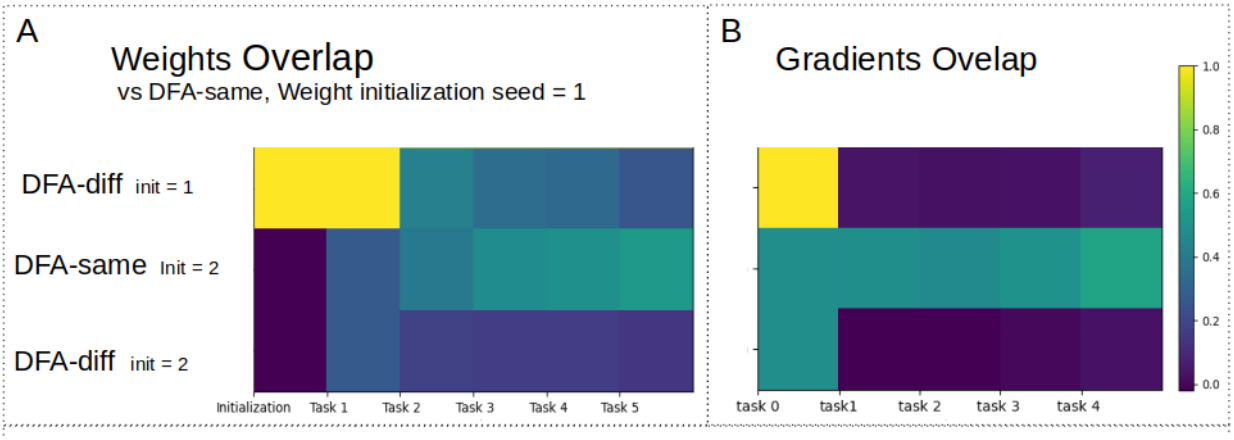

Figure 6: **A**: Dot product of the weights at the end of training on the different tasks (averaged over the first two layers, Domain-IL architecture structure) of two networks trained with DFA-same or DFA-diff in contrast with a reference network trained with DFA-same. **Weight Alignment** can be appreciated mostly by looking at the second row, the case in which the networks are initialized from different seeds (seed 1 for the reference network and seed 2 for the network in the second row): the weights at first do not overlap, but thanks to the fact that the networks are trained with the same feedback matrix, the weights overlap more and more. **B**: Overlap of gradients averaged over two different checkpoints during the training on the same dataset (splitFMNIST, Task-IL architecture structure) of two networks trained with DFA-same or DFA-diff in contrast with a reference network trained with DFA-same. **Gradient Alignment** is more visible in the first and the third row: the gradients in panel B become orthogonal (overlap equal to 0) starting from the second task, as soon as the feedback matrix changes, irrespective of the fact that the networks are initialized in the same way (first row) or if the networks are initialized differently (third row).

B DFA-same versus BP and BP-ablated in the long run

In order to answer the question, "Is DFA mitigation for catastrophic forgetting solely due to the small learning rate in the first two layers?" we propose a deeper comparison with BP ablated in the context of the permuted dataset. The same results apply to the split dataset. In fig. 7 below, we display the accuracy of all the tasks during the CL pipeline (training one task at a time and always monitoring the accuracy of the other tasks; in the case of Task-IL scenario, either for training and for testing one task, the corresponding output layer is used). We compare the case in which BP ablated has the most similar AF to DFA-same and we find that DFA has the tendency of forgetting more than BP and BP ablated after few stages but is more stable in the long run, for example after 6 tasks in the Task-IL and after 1 or 9 tasks in the Domain-IL scenario; the black and the grey lines in the plot of DFA-same visually show the point where the advantage starts. We conclude that a small learning rate in the first two layers is not always enough to reproduce DFA-same stability and this advantage in the long run might be due to the alignment of the weights.

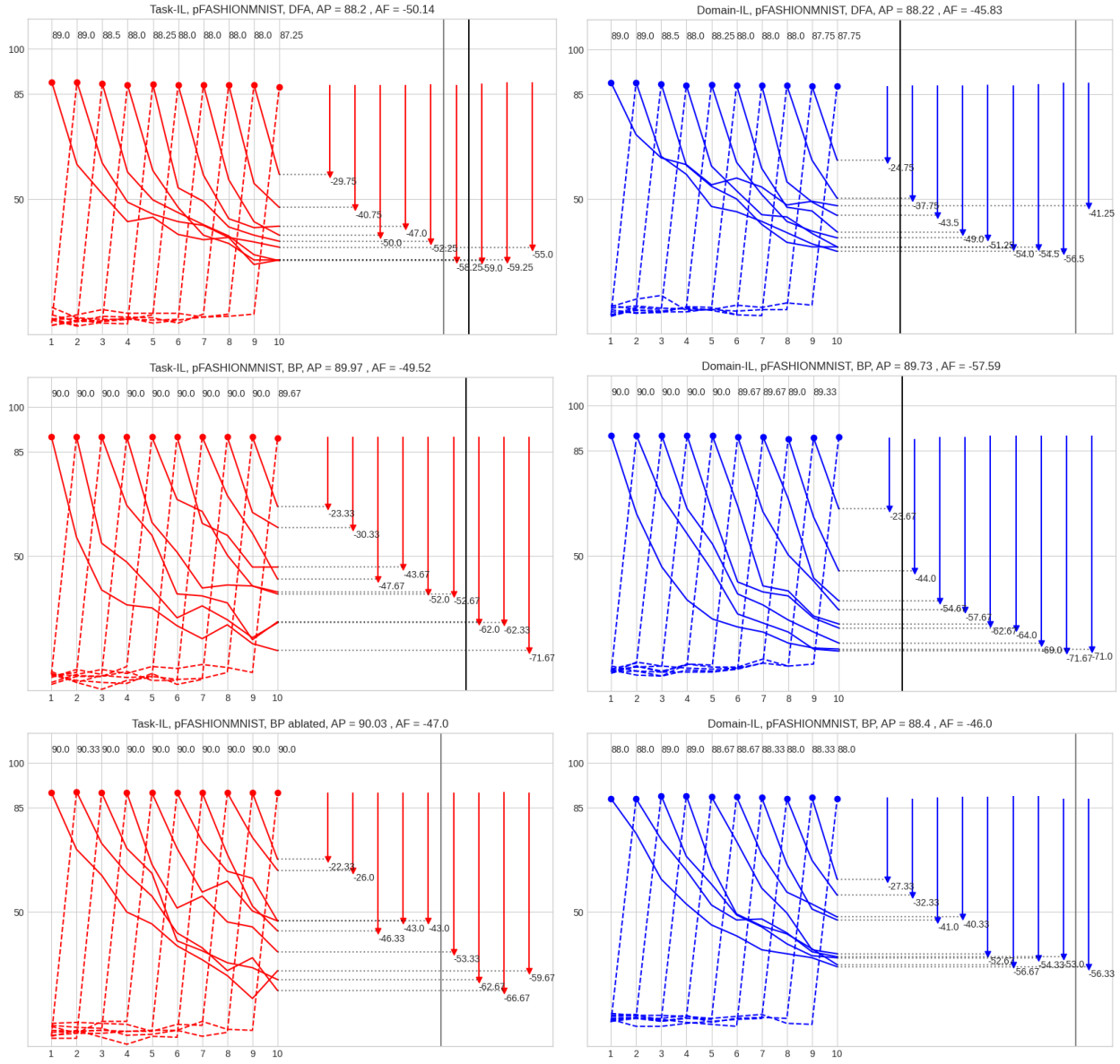

Figure 7: Accuracy interpolation lines along the stages. On the left (red lines) we show the results for the Task-IL scenario, we can notice that even though DFA-same has higher average forgetting than BP and BP-ablated, it has lower forgetting for the first three (vs BP) or four tasks (vs BP ablated): DFA-same retains the accuracy better than BP and BP-ablated in the long run, while it forgets more in the shorter run. This is also true in the Domain-IL scenario (blue lines, on the right) in which DFA has better average forgetting than BP and worse compared to BP-ablated.

C Results evaluated on MNIST

The plots displaying the results of the methods applied to the MNIST datasets in the Task-IL scenario can be found in fig. 8. The points in the circle indicates the Split-MNIST, otherwise they refer to the results on the Permuted-MNIST.

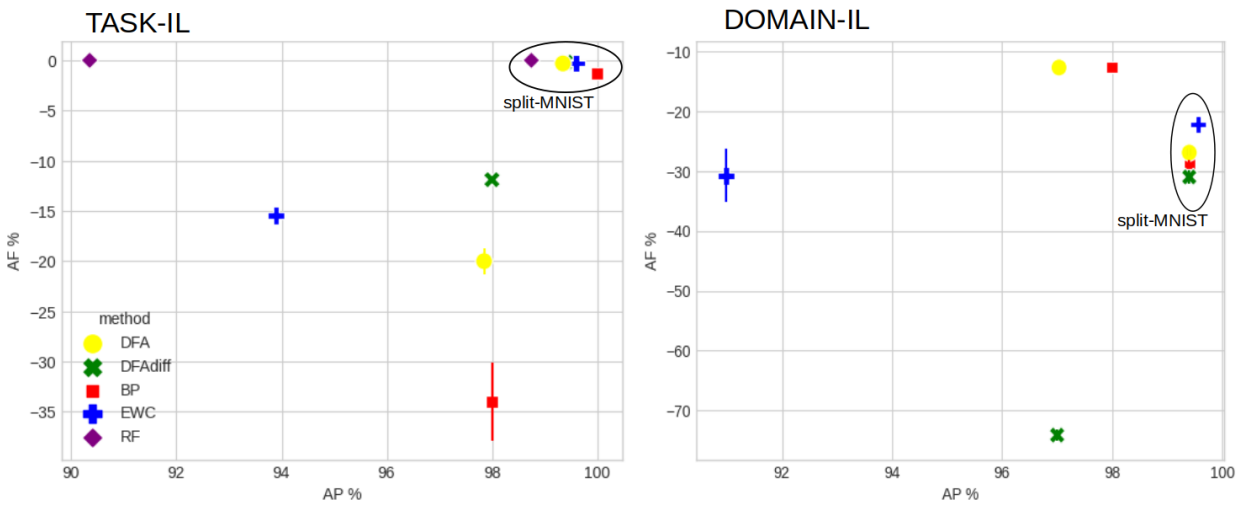

Figure 8: We can notice that the results are consistent with the results on F-MNIST: In the Task-IL split-MNIST DFA lays in an intermediate position between RF and BP, slightly behind EWC; In the permuted Task-IL, DFA-diff has less forgetting than all the other methods while achieving the same AP of BP. In Domain-IL (panel on the right), DFA-same is comparable or equal in AF to BP while DFA-diff has the worst performances; EWC is in an intermediate level between DFA-same and DFA-diff on the permuted dataset while is superior to DFA on the split datasets.

D Random Seed impact on the accuracy of the firstly learned task

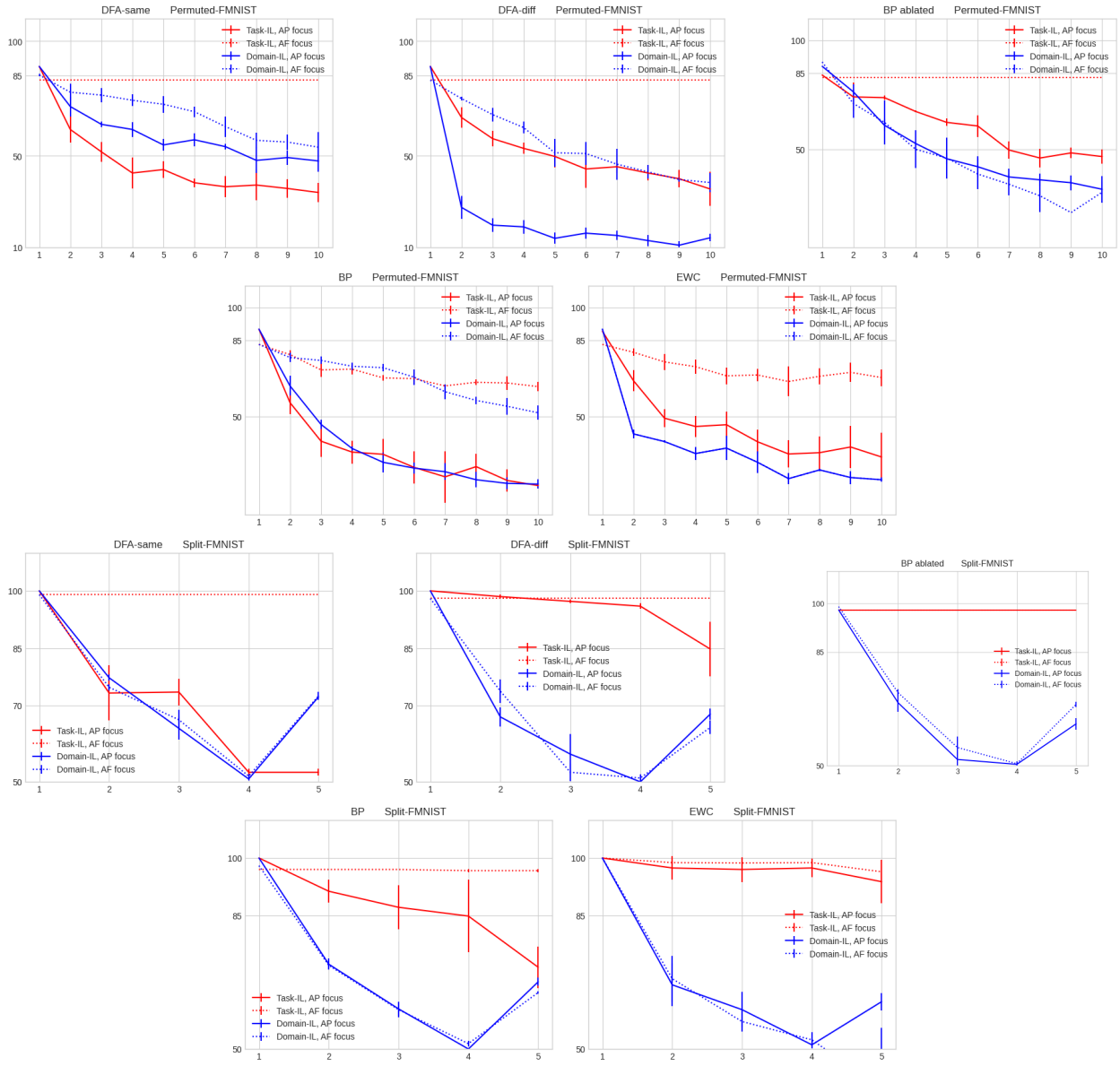

Figure 9: Complete illustration of forgetting the first learned task for all methods (Accuracy % on the y-axis and stage number on the x-axis). The different figures display one method at a time in the following order: DFA-same, DFA-diff, BP-ablated, BP, EWC; permuted FMNIST above and split FMNIST below. The dotted lines are the ones optimized for minimizing forgetting. In EWC permuted FMNIST, the blue dotted line coincides with the blue solid line.

E Similarity Measures of datasets

In fig. 1, panel B, we analyze the similarity of the datasets arising from the division into the different tasks. We take in the rows all the images of one class in the dataset corresponding to task1 (and their pixels in the columns) and we compare it to the matrix of images of the same class in the next dataset. We average over all the couples of datasets and over the different classes. The similarity measures we apply (described below) measure the similarity from a merely geometrical perspective and we find that in this point of the tasks have classes more similar to each other in the permuted dataset. By construction, the reshuffle of the pixels does not change the mean and the standard deviation of the distributions of the pixels. Nevertheless, the split datasets is easier because it has only two classes and in the Task-IL learning the difference in distribution can be an advantage. In fact, the average forgetting in the Task-IL scenario are overall much smaller than the average forgetting on the permuted dataset. On the other hand, the permuted dataset has only slightly more forgetting in the Domain-IL scenario in the case in which the models are optimized for performance.

we use the following similarity measures:

1. Cosine similarity: Dot product of the matrices reshaped into vectors and subsequently normalized by the L2 norm of the two vectors.
2. Canonical Correlation Analysis (CCA) (Hardoon et al., 2005): Similarity measure which is invariant to affine transformations (any linear invertible transformation including scaling, rotations, translations).
3. Centered Kernel Alignment (CKA) (Kornblith et al., 2019): Similarity measure which is invariant to orthogonal transformations, which is a subset of the invertible linear transformations.

CCA is 1 when the two matrices are linearly invariant (they are linked by any affine transformation) and 0 when they are not; CKA is 1 in the case that the two matrices are orthogonally invariant (can be linked by a rotation).

With the plot in fig. 1 we show that the permuted dataset is composed of images with distributional similarity among the tasks. This holds also from the point of view of the representations inside the network of the different tasks. We report in table 2 the similarity of the activation matrices (input images in the rows, and number of neurons in the columns). In the permuted dataset there is a larger overlap (especially in the case with shared output) than the disjoint datasets.

F Analysis of the representations

Since the training is driven by both the architecture (weights, activation functions) and the data distribution, it can useful to inspect how the activations of the network in different stages vary during the training on different tasks.

In Neuroscience, similarity measures of the activations of neurons have proven to be a valid tool for computing differences in the activation patterns of neurons. These methods have been applied to ANNs to compare the internal representations of the networks among layers, epochs of training, different initialization seeds, different training algorithms, training architectures, and datasets.

We showed that the gradients and weights align in two networks with different weight initialization and same feedback matrix (fig. 4 and Appendix appendix A). In this part, we provide an analysis of the representations overlap. In table 2 we report the dot product of the differences in activations of the fist two layers obtained feeding the test images of two subsequent datasets into the network in the same stage of training, averaged for all couples of adjacent datasets and stages of training. It is evident that in the split datasets the activations of subsequent tasks do not overlap at all; while in the permuted dataset some overlap is present.

Table 2: The dot product represents the overlap of the activations (0: no overlap, 1: complete overlap).

		Task-IL Dot product of changes	Domain-IL overlaps of different datasets
Split	BP	0	0
	DFA-same	0	0
	DFA-diff	0	0
	EWC	0	0
Permuted	BP	0.045	0.87
	DFA-same	0.08	0.96
	DFA-diff	0.09	0.97
	EWC	0.06	0.92