Online Continual Learning through Target Regularization

Francesco Lässig, Pau Vilimelis Aceituno, Martino Sorbaro, Benjamin F. Grewe
Institute of Neuroinformatics, University of Zürich and ETH Zürich
flaessig@ethz.ch, begrewe@ethz.ch

I. Introduction

The continual learning (CL) problem represents a major challenge in deep learning [1]. Most commonly used are weight regularization methods that guide weight updates to avoid interference with previous tasks. These range from weight constraints embedded in the loss function [2], [3] to the extreme case of training distinct subnetworks. The latter can be achieved by freezing parameters used for previous tasks, choosing task-specific weights by neuronal pruning, or by applying pre-defined masks to select a task specific subnetwork [4], [5]. Although weight regularization techniques greatly improve CL performance, they usually require task boundaries, which is biologically implausible and limits the range of possible learning scenarios.

Following R. French's early idea about alleviating forgetting by reducing representational overlap [6], we propose an approach to online CL that restricts learning to sparse neuronal representations that are dynamically inferred for each data point. To learn, our approach does not use backpropagation (BP), but instead builds on a bio-inspired form of hierarchical credit assignment known as Deep Feedback Control (DFC) [7]. In contrast to standard deep learning, the DFC network is continuous in time and relies on a dynamic top-down feedback controller. During learning, the controller drives neurons to specific target activations until the network output matches its target. The contributions of the controller to indiviudual neuron activations determine their weight updates. To avoid forgetting, we modulate the network learning dynamics to converge on sparse target representations to which we restrict the feedforward weight updates. We utilize a simple sparsity mechanism which ensures that only highly selective neurons remain active as driven by the input or feedback. Having learned to represent a data point with sparse activities that minimize the loss, we 'engrave' the structure of this representation into the network by learning lateral inhibitory connections within each hidden layer. This allows subsequent tasks to be represented in a way that is consistent with the structure used to represent previous tasks. In contrast to the feedforward weights, we restrict recurrent weight updates to inactive neurons that are excluded from the target representation (Fig. 1). In the next section we provide further implementation details on how we modified the DFC learning dynamics to integrate the two major factors required for CL – sparsity and lateral (recurrent) inhibition. We term this combination *sparse-recurrent* DFC.

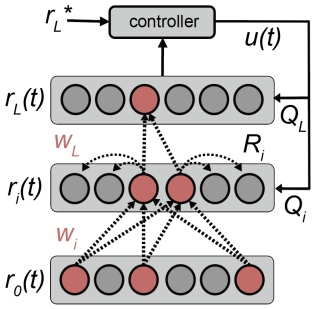

Fig. 1. Schematic of DFC network and top down feedback controller. Learning is based on a dynamic process during which all hidden neurons integrate feedforward and feedback signals until the network converges to a sparse target representation minimizing the loss. Weight updates (dashed lines) of W_i are restricted to active neurons comprising the target (red). Recurrent weights R_i of inactive neurons are updated via an anti-hebbian learning rule, feedback weights Q_i are fixed.

II. Target Regularization with Sparsity and Lateral Inhibition

A. Learning Dynamics

During training, the neuronal dynamics within the standard DFC network [8] can be described by a differential equation that takes into account the feedforward inputs v_i^{ff} as well as the feedback control signal v_i^{fb} according to

$$\begin{aligned} \tau_v \ \dot{v}_i(t) &= -v_i(t) + \quad v_i^{\text{ff}}(t) \quad + \ v_i^{\text{fb}}(t) \\ &= -v_i(t) + W_i \ \phi\big(v_{i-1}(t)\big) + Q_i \ u(t) \end{aligned} \tag{1}$$

where the pre-non-linearity neuron activations in layer i at time t are denoted by $v_i(t)$, and the incoming weights by W_i. ϕ refers to the activation function while the neuron output is given by $r_i = \phi(v_i(t))$. The feedback signal $u(t)$ is computed as described in [8] by summing the integral and proportional parts of the network output error. The feedback signal $u(t)$ is then fed back to each neuron of the network via the feedback weights Q_i. During learning, the feedforward network and the feedback controller constitute a recurrent dynamical system that converges to a final target at which the neuron activations $v_{i,\text{ss}}$ minimize the output error and stabilize the feedback signal $u(t)$. For updating the feedforward weights at the stable (converged) state (ss) each neuron's target activation is compared to its initial feedforward activation according to

$$\Delta W_{i,\text{ss}} = \eta \ \left(r_{i,\text{ss}} - \phi(v_{i,\text{ss}}^{\text{ff}})\right) r_{i-1,\text{ss}}^T \tag{2}$$

where $r_{i-1,\text{ss}}^T$ is the pre-synaptic target activity with controller feedback. $r_{i,\text{ss}}$ is the target activity of the neuron with feedback and $\phi(v_{i,\text{ss}}^{\text{ff}})$ is the postsynaptic neuron activity without feedback. Although feedback weights Q_i can be learned [7], [8], we simplify the learning of the feedback

pathway and re-initialize Q_i as the Jacobian of the loss with respect to the neuron activations for every data point.

B. Dynamic Sparsity

To gradually modulate the network learning dynamics towards sparse targets, we add a winner-take-all (WTA) mechanism on top of the existing DFC network. At each time step t we set a small fraction $s_i(t)$ of neurons to be zero. We then increase $s_i(t)$ dynamically until the desired sparsity for the stable state $s_{i,\text{ss}}$, which is a hyperparameter fixed for each layer. Although this mechanism is sufficient to reach the prespecified target sparsity, it does not yet guarantee that sparse representations will be consistent across tasks. To ensure a shared representational structure across tasks, we introduce a second mechanism to 'engrave' structure of activity patterns into the network.

C. Lateral Inhibition

We imprint information about mutual exclusivity between populations of neurons into the network using lateral recurrent connections. Because we want the recurrent weights to strongly influence which combinations of neurons are allowed to comprise the target, as opposed to incrementally affecting their activities, we introduce these as multiplicative weights between 0 and 1, similar to 'forget' gates used in LSTMs [9]. We then calculate the neuron feedforward activity before the nonlinearity according to

$$v_i^{\text{ff}}(t) = W_i\,\phi\big(v_{i-1}(t)\big) \odot \sigma\big(R_i\,|r_i(t-1)|\big) \tag{3}$$

where R_i refers to the recurrent weight matrix in the i-th layer. At convergence, we only learn the recurrent inhibitory weights for all inactive neurons according to a simple anti-Hebbian update rule

$$\Delta R_i = -\eta\,|\phi(v_{i,\text{ss}}^{\text{ff}})|\,|r_{i,\text{ss}}|^T \tag{4}$$

where $r_{i,\text{ss}}^T$ are the target activities of the presynaptic neurons in the same layer. Since we used a $\tanh$ activation function in combination with multiplicative weights we use the absolute activity values for updating the recurrent weights. For inactive neurons we only update incoming recurrent weights. For active neurons that comprise the target representation we only update the incoming feedforward weights. Fig. 1 (dashed lines) summarizes the weight updates. We next show that the specific combination of feedback, recurrent dynamics and sparsity represents a new, competitive CL approach across a wide range of learning rates (LRs).

III. Experiments

To test the CL capabilities of our approach, we next train sparse-recurrent DFC on the split-MNIST dataset according to the domain incremental learning paradigm as outlined by Vandeven 2019 [10]. Previous works [2], [10] evaluate models at fixed LRs for a fixed number of epochs. We consider this as problematic because LR can be seen as a proxy for how much a network learns, and there is an inherent trade-off between learning the current task well and forgetting previous

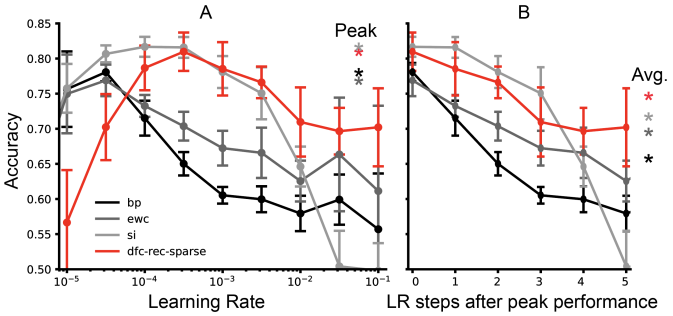

Fig. 2. LR-sweep evaluation of domain-IL split-MNIST performance of BP, EWC, SI and DFC-sparse-rec. **A**: Accuracy of network at the end of training on the whole test set for every LR. Error bars were produced by running every model on every LR for 5 random seeds. Stars indicate peak performance of each method. **B**: Peak-aligned final test set accuracy curves across LRs. The curves shown here correspond to the sections found in **A** where the starting point for each model was chosen as the peak performance. Thus this panel shows the peak performance for each model as well as the following 5 values, regardless of absolute LR. Stars indicate the average of the performance decay according to 6 consecutive LR values.

tasks. Low LRs generally delay forgetting, while at the same time slowing down learning the current task. Evaluating CL algorithms at a single LR is not only problematic because it doesn't account for different optimal LRs, but it also fails to capture how robust a model is to more learning, beyond its optimum LR. To overcome this issue, we compare our approach against the most common CL methods across a wide range of LRs (Fig. 2). These include classical CL methods such as Synaptic Intelligence (SI), Elastic Weight Consolidation (EWC), as well as standard BP as baseline. Fig. 2A shows that sparse-recurrent DFC generally outperforms both BP and EWC for most LRs. The asymmetric shape of the performance curves can be explained by the fact that very small LRs (left of the peak) generally prevent learning while high LRs (right of peak) can lead to substantial catastrophic forgetting. Moreover, the individual CL performance profiles confirm our initial intuition that choosing a single LR to compare CL methods might lead to overestimating one method over another. If aligned to peak performance (Fig. 2B) sparse-recurrent DFC outperforms BP and EWC for every LR. However, SI exhibits a slightly higher peak performance while sparse-recurrent DFC shows a slower decay and better performance at higher LRs. To cover and compare these effects we evaluate CL performance either by comparing peak performances or by taking the performance average of a contiguous window of LRs once peak performance has been reached (Fig. 2, stars). The latter evaluation metric has the desired property of favoring higher peak performance followed by slower decays while being indifferent to the optimal LR. Overall, we conclude that target regularization in our DFC framework represents a competitive CL method across a large range of LRs. In the next section we will investigate in more detail the effect of the controller feedback signal to facilitate CL.

A. *Integration feedback (error) signaling facilitates CL*

A major difference between standard BP and DFC is that in DFC, the activity of each neuron during training reflects feedforward as well as feedback (error) signals coming from the top-down controller. As a result, sparse target representations are specific to both input and output, with data points exhibiting larger overlaps in target representations if these have similar features *or* the same label. Fig. 3A shows that the CL performance is improved across a wide range of LRs if we take into account feedback signals when selecting the remaining sparse target representation. We conclude that the feedback signals help to choose the right neuron populations as sparse targets. We next investigate the combination of sparsity and recurrence to enable CL in the DFC framework.

B. *Sparsity and lateral inhibition are required for CL*

To investigate whether both sparsity and recurrent weights are necessary for CL, we compare the accuracy of sparse-recurrent DFC against standard DFC, sparse DFC and recurrent DFC. Fig. 3B shows that neither sparsity nor recurrent connections alone significantly alter CL performance across LRs. However, the combination of the two leads to better performance across most LRs. Comparing peak accuracy (Fig. 3B, stars denoted by 'Peak') and the average accuracy over a contiguous post-peak window of six LR steps (Fig. 3B, stars denoted by 'Avg.') confirms that both sparsity and recurrence are required for performance gains at optimal LRs as well as improvements in post-peak performance.

C. *Reduction in overlap correlates with performance gains*

Next, we investigate if the combination of sparsity and lateral inhibition avoids catastrophic forgetting by reducing representational overlap. We therefore compute the reduction in overlap (i.e. separation) of active neurons in the last hidden layer between representations of all pairs of digits at the end of training. We distinguish between intra-label separation (MNIST digits with the same label) and inter-label separation (digits with a different labels). We compute representational separation between digits as

$$s(d_1, d_2) = 1 - \frac{a_l^{d_1} \cdot a_l^{d_2}}{\|a_l^{d_1}\| \|a_l^{d_2}\|}; \qquad a_l^d = \sum_{j=1}^{n} |r_{l,j}^d| \tag{5}$$

where $r_{l,j}^d$ represents the activations in layer l elicited by the j'th sample of digit d. Fig. 3C shows the averages of inter- and intra-label representational separations for DFC variants. Interestingly, sparse DFC does not yield significantly higher accuracies compared to standard DFC or BP, suggesting that overall increases in representational separation do not account for performance improvements that we observed in Fig. 3B. To resolve this issue, we define a new measure, that we term information distance, as the inter divided by the intra label separation. Fig. 3D shows that this information distance over a wide range of LRs. For the LRs where sparse-recurrent DFC yields higher information distance, we also observe better CL performance (compare to Fig. 3B), suggesting that the relative

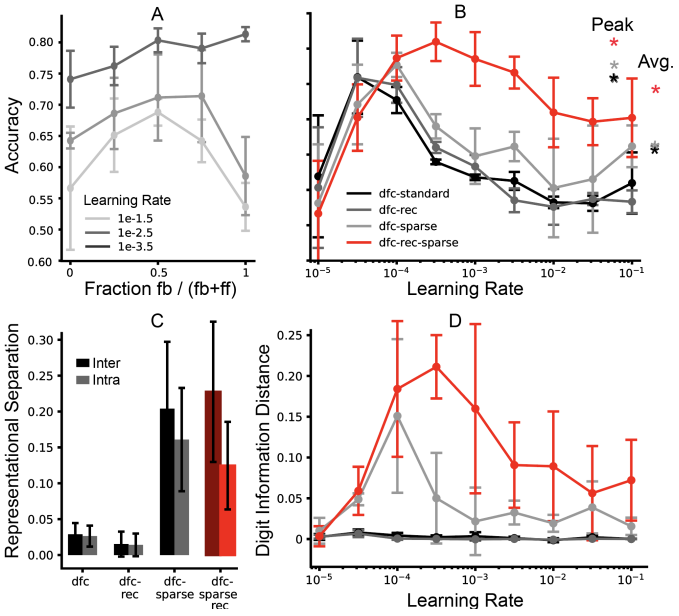

Fig. 3. **A** Effect of feedforward and feedback activity used in selecting the sparsified population for different LRs. The x-axis represents the fraction measuring the impact of feedback activity on the selection of neurons to be sparsified. A value of 0 means only feedforward activity is considered, a value of 1 means only feedback is taken into consideration, and 0.5 corresponds to an equal mix of the two activites. **B** Cross-LR evaluation for all DFC variants. The plot reflects the overall performance for all digits at the end of training. **C.** Inter- and intra-label separations for DFC variants after all five tasks have been learned. Intra-label separations are calculated for all digit pairs with same label, inter-label separations for all pairs of digits with different labels. **D** Information distance calculated as the inter-digit separation divided by intra-digit separation at the end of training across a wide range of LRs.

degree of digit representational overlap can explain the CL performance profile that we observe for sparse-recurrent DFC.

D. *Recurrent weights constrain learning across tasks*

In the case of domain-IL, the network has to learn a representation of its input in the final hidden layer for each task which is linearly separable by its readout weights. One possible way to prevent forgetting is to ensure two things. **Requirement 1**: The hyperplane separating representations of different labels (implemented in the network by the readout layer) needs to stay the same, or similar to the old one. **Requirement 2**: Data points represented in the final hidden layer need to stay on the same side of the classification hyperplane that was initially learned. We measured feedforward and target activations (including effects of controller and recurrent connections) of the final hidden layer of the network to test whether recurrent weights help to achieve this.

Regarding requirement 1, Fig. 4B shows that, if we classify targets at the start of training of a new task according to the previously learned separation boundary, DFC-sparse-rec consistently yields higher classification accuracies than DFC-sparse. This suggests that recurrent weights regularize new targets such that they align with the previously learned boundary. This idea is illustrated in Fig. 4A, where task 2 targets are separated by the same hyperplane that divides

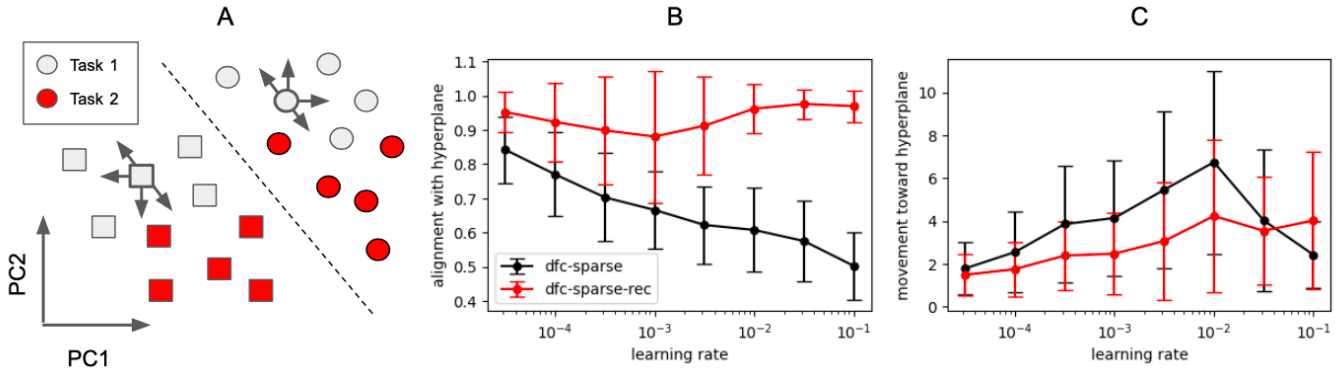

Fig. 4. Effects of recurrent weights on targets and feedforward activations during learning. **A**: Schematic of task 1 and 2 representations with respect to the hyperplane (dashed line) dividing task 1 targets (grey) according to their label. This diagram illustrates two things: First, the new target representations align with the previously learned hyperplane in terms of label separation (supported by **B**). In other words, the hyperplane that separates task 1 targets also separates task 2 targets. Second, task 1 representations generally move less towards the separating hyperplane as subsequent tasks are learned (supported by **C**). This is represented by the arrows. **B**: Fraction of initial target representations ($v_{i,\mathrm{ss}}$) of new task that are correctly separated acccording to the previously learned hyperplane. **C**: Movement of feedforward activations ($v^{\mathrm{ff}}_{i,\mathrm{ss}}$) towards hyperplane after learning subsequent tasks, normalized by movement in all directions.

task 1 targets. Regarding requirement 2, we measured the direction of movement of feedforward activations from when they were first learned to the end of training. We quantify how much datapoints move towards the initially learned separation boundary. Fig. 4C shows movement towards the hyperplane normalized by movement in other directions. We can see that DFC-sparse-rec reduces this movement compared to DFC-sparse, although error bars are large.

IV. Discussion

Sparse-recurrent DFC yields CL performance improvements beyond what sparsity and recurrent connections achieve individually. The necessity of sparsity supports our hypothesis that catastrophic forgetting is alleviated by establishing a separation of active neuron populations. However, the separation of active neuron populations between representations of any two digits alone does not explain the performance improvement, since sparse DFC yields overall separation levels comparable to sparse-recurrent DFC (Fig. 3C). Instead, the decisive measure seems to be how separated neuron populations representing distinct outputs are. Overall, this suggests that sparse-recurrent DFC improves domain-IL split-MNIST performance by creating two (partially) separated neuron populations in the last hidden layer, each of which is selective for a given label (digit parity), but not for a specific task (which pair of digits). The necessity of lateral inhibition can be explained by its effect of aligning new targets to old separation boundaries, thus reusing the structure established by previous tasks. This structure that is imprinted onto recurrent weights in early tasks and reused in later tasks consists of information about which neurons can fire at the same time, and which are mutually exclusive. Further, this structure constrains the movement of previously learned data points towards the hyperplane, thus preserving representations that are compatible with their initially learned separation boundary. The exact mechanism by which this happens still needs to be investigated.

One drawback of our approach compared to standard machine learning methods is that it is less computationally efficient due to the need to approximate differential equations of the network dynamics. However, we are optimistic that the same principles that enabled sparse-recurrent DFC to improve upon other strategies are translatable to more classical, GPU-friendly implementations. Alternatively, a neuromorphic system implementation that physically emulates neuron dynamics could also solve this problem.

Our results show that DFC-sparse-rec not only performs better than standard DFC and BP on split-MNIST CL tasks, but also better than EWC and arguably on par with SI. Moreover, since sparse-recurrent DFC relies on principles that are distinct from the ones used in EWC and SI, it could potentially be combined with additional loss terms to yield even better CL performance. On a similar note, our method does not require any specific action at the task boundary, whereas EWC and SI update their loss term at the end of every task. This renders our method both more biologically plausible and potentially more versatile. Not requiring task boundaries could be especially helpful in online learning scenarios where changes in distributions of the input data are not known.

With this work we show that we can match and in some cases even exceed the performance of existing CL approaches by using principles of neural computation inspired by biology. From a machine learning perspective this is relevant because we are using a different set of guiding principles than existing approaches, which opens the possibility of combining ideas from both domains for even better results. Although the current implementation of sparse-recurrent DFC is less efficient compared to standard learning algorithms when run on GPUs, we believe that future work could translate our approach into a much more efficient implementation. From a neuroscientific perspective, our findings allow experimenters to derive new hypotheses about how the brain might avoid catastrophic forgetting.

REFERENCES

[1] Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, and Z. Kira, "Re-evaluating continual learning scenarios: A categorization and case for strong baselines," 10 2018. [Online]. Available: http://arxiv.org/abs/1810.12488

[2] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," 12 2016. [Online]. Available: http://arxiv.org/abs/1612.00796

[3] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," 3 2017. [Online]. Available: http://arxiv.org/abs/1703.04200

[4] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[5] S. Golkar, M. Kagan, and K. Cho, "Continual learning via neural pruning," 3 2019. [Online]. Available: http://arxiv.org/abs/1903.04476

[6] R. M. French, "Using semi-distributed representations to overcome catastrophic forgetting in connectionlst networks," 1993. [Online]. Available: www.aaai.org

[7] A. Meulemans, M. T. Farinha, J. G. Ordóñez, P. V. Aceituno, J. Sacramento, and B. F. Grewe, "Credit assignment in neural networks through deep feedback control," 6 2021. [Online]. Available: http://arxiv.org/abs/2106.07887

[8] A. Meulemans, M. T. Farinha, M. R. Cervera, J. Sacramento, and B. F. Grewe, "Minimizing control for credit assignment with strong feedback," 4 2022. [Online]. Available: http://arxiv.org/abs/2204.07249

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] G. M. van de Ven and A. S. Tolias, "Three scenarios for continual learning," 4 2019. [Online]. Available: http://arxiv.org/abs/1904.07734