Differentiable Hebbian Plasticity for Continual Learning

Vithursan Thangarasa [1 2 *] **Thomas Miconi** [3] **Graham W. Taylor** [1 2 †]

Abstract

Catastrophic forgetting poses a grand challenge for continual learning systems, which prevents neural networks from protecting old knowledge while learning new tasks sequentially. We propose a Differentiable Hebbian Plasticity (DHP) Softmax layer which adds a fast learning plastic component to the slow weights of the softmax output layer. The DHP Softmax behaves as a compressed episodic memory that reactivates existing memory traces, while creating new ones. We demonstrate the flexibility of our model by combining it with existing well-known consolidation methods to prevent catastrophic forgetting. We evaluate our approach on the Permuted MNIST and Split MNIST benchmarks, and introduce Imbalanced Permuted MNIST — a dataset that combines the challenges of class imbalance and concept drift. Our model requires no additional hyperparameters and outperforms comparable baselines by reducing forgetting.

1. Introduction

A key aspect of human intelligence is the *ability to continually adapt and learn* in dynamic environments, a characteristic which is challenging to embed into artificial intelligence. Recent advances in machine learning (ML) have shown tremendous improvements in various problems, by learning to solve one complex task very well, through extensive training on large datasets with millions of training examples or more. Most of the ML models that we use during deployment assume that the real-world is stationary, where in fact it is non-stationary and the distribution of acquired data changes over time. Therefore, after learning is complete, and these models are fine-tuned with new data, performance degrades with respect to the original data. This phenomenon known as *catastrophic forgetting* or *catastrophic interference* (McCloskey & Cohen, 1989; French, 1999) serves to be a crucial problem for deep neural networks (DNNs) that are tasked with *continual learning* (Ring, 1994) or *lifelong learning* (Thrun & Mitchell, 1995). In this learning paradigm, the goal is to adapt and learn consecutive tasks without forgetting how to perform previously learned tasks. Some of the real-world applications that typically require this kind of learning include perception for autonomous vehicles, recommender systems, fraud detection, etc.

In most supervised learning methods, DNN architectures require independent and identically distributed (iid) samples from a stationary training distribution. However, for ML systems that require continual learning in the real-world, the iid assumption is easily violated when: (1) There is concept drift or class imbalance in the training data distribution. (2) Data representing all scenarios in which the learner is expected to perform are not initially available. In such situations, DNNs face the "stability-plasticity dilemma" (Carpenter & Grossberg, 1987; Abraham & Robins, 2005). This presents a continual learning challenge for models that need to balance plasticity (integrate new knowledge) and stability (preserve existing knowledge).

Two major theories have been proposed to explain a human's ability to perform continual learning. The first theory is inspired by synaptic consolidation in the mammalian neocortex (Benna & Fusi, 2016) where a subset of synapses are rendered less plastic and therefore preserved for a longer timescale. The second theory is the complementary learning systems (CLS) theory (McClelland et al., 1995; O'Reilly et al., 2014; Kumaran et al., 2016), which suggests that humans extract high-level structural information and store it in a different brain area while retaining episodic memories.

Here, we extend the work on differentiable plasticity (Miconi, 2016; Miconi et al., 2018) to a continual learning setting and develop a model that is capable of adapting quickly to changing environments as well as consolidating previous knowledge by selectively adjusting the plasticity of synapses. We modify the traditional softmax layer and propose to augment the slow weights with a set of plastic weights implemented using Differentiable Hebbian Plasticity (DHP). The model's slow weights learn deep representations of data and the fast weights implemented with DHP

*Work done during an internship at Uber AI. † Work done while at Google Brain. [1]School of Engineering, University of Guelph, Canada [2]Vector Institute for Artificial Intelligence, Canada [3]Uber AI, USA. Correspondence to: Vithursan Thangarasa <vthangar@uoguelph.ca>.

Proceedings of the 1^{st} Adaptive & Multitask Learning Workshop, Long Beach, California, 2019. Copyright 2019 by the author(s).

learn to quickly "auto-associate" the class labels to representations. We also demonstrate the flexibility of our model by combining it with recent task-specific synaptic consolidation based methods to overcoming catastrophic forgetting such as elastic weight consolidation (Kirkpatrick et al., 2017; Schwarz et al., 2018), synaptic intelligence (Zenke et al., 2017) and memory aware synapses (Aljundi et al., 2018). Our model unifies core concepts from Hebbian plasticity, synaptic consolidation and CLS theory to enable rapid adaptation to new unseen data, while consolidating synapses and leveraging compressed episodic memories to remember previous knowledge and mitigate catastrophic forgetting.

2. Relevant Work

Plastic Neural Networks: One of the major theories that have been proposed to explain a human's ability to learn continually is Hebbian learning (Hebb, 1949), which suggests that learning and memory are attributed to weight plasticity, that is, the modification of the strength of existing synapses according to variants of Hebb's rule (Paulsen & Sejnowski, 2000; Song et al., 2000; Oja, 2008).

Recent approaches in the meta-learning literature have shown that we can incorporate fast weights into a neural network (Munkhdalai & Trischler, 2018; Rae et al., 2018). Munkhdalai & Trischler (2018) augmented fully-connected (FC) layers preceding the softmax with a matrix of fast weights. Here, the fast weights were implemented with *non-trainable* Hebbian learning-based associative memory. Rae et al. (2018) proposed a softmax layer that can improve learning of rare classes by interpolating between Hebbian updates and stochastic gradient descent (SGD) updates on the output layer using an arbitrarily engineered scheduling scheme. Miconi et al. (2018) proposed differentiable plasticity, which uses SGD to optimize the plasticity of each synaptic connection composed of a slow weight and a plastic (fast) weight. Although this approach served to be a powerful new method for training neural networks, it was mainly demonstrated on RNNs for solving simple tasks.

Overcoming Catastrophic Forgetting: This work leverages two biologically inspired strategies to overcome the catastrophic forgetting problem: 1) *Task-specific Synaptic Consolidation* — Protecting old knowledge by dynamically adjusting the synaptic strengths to consolidate and retain memories. 2) *CLS Theory* — A dual memory system where, structural knowledge is acquired through slow learning via the neocortex and rapid learning via the hippocampus.

There have been several notable works inspired by task-specific synaptic consolidation for overcoming catastrophic forgetting (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018). All of these approaches propose a method to estimate the importance of each parameter or synapse, Ω_k, where the least plastic synapses can retain memories for a long time and the more plastic synapses are considered less important. The Ω_k and network parameters θ_k are updated online or after learning task T_n. Therefore, when learning new task T_n, a regularizer is added to the original loss function $\mathcal{L}^n(\theta)$, so that we dynamically adjust the plasticity w.r.t. Ω_k and prevent any changes to the important parameters of previously learned tasks:

$$\tilde{\mathcal{L}}^n(\theta) = \mathcal{L}^n(\theta) + \underbrace{\lambda \sum_k \Omega_k (\theta_k^n - \theta_k^{n-1})^2}_{\text{regularizer}} \tag{1}$$

where θ_k^{n-1} are the learned network parameters after training on the previous $n-1$ tasks and λ is a hyperparameter for the regularizer to control the amount of forgetting.

In Elastic Weight Consolidation (EWC), Kirkpatrick et al. (2017) use the diagonal values of an approximated Fisher information matrix for Ω_k, and it is computed offline after training on a task is completed. Schwarz et al. (2018) proposed an online variant of EWC to improve scalability by ensuring the computational cost of the regularization term does not grow with the number of tasks. Zenke et al. (2017) proposed an online method called Synaptic Intelligence (SI) for computing the parameter importance where, Ω_k is the cumulative change in individual synapses over the entire training trajectory on a given task. Memory Aware Synapses (MAS) from Aljundi et al. (2018) measures Ω_k by the sensitivity of the learned function to a perturbation in the parameters and use the cumulative change in individual synapses on the squared L2-norm of the penultimate layer.

There have been numerous approaches based on CLS principles involving pseudo-rehersal (Robins, 1995; Ans et al., 2004; Atkinson et al., 2018), episodic replay (Lopez-Paz & Ranzato, 2017; Li & Hoiem, 2018) and generative replay (Shin et al., 2017; Wu et al., 2018). However, in our work, we are primarily interested in neuroplasticity techniques inspired from CLS theory for representing memories.

Hinton & Plaut (1987) showed how each synaptic connection can be composed of a fixed weight where slow learning stores long-term knowledge and a fast weight for temporary associative memory. Recent research in this vein has included replacing soft attention mechanism with fast weights in RNNs (Ba et al., 2016), the Hebbian Softmax layer (Rae et al., 2018), augmenting the FC layer with a fast weights matrix (Munkhdalai & Trischler, 2018), differentiable plasticity (Miconi et al., 2018) and neuromodulated differentiable plasticity (Miconi et al., 2019). However, all of these methods were focused on rapid learning on simple tasks or meta-learning over a distribution of tasks. Furthermore, they did not examine learning a large number of new tasks while, alleviating catastrophic forgetting in continual learning.

3. Model

In our model, each synaptic connection in the softmax layer has two weights: 1) The slow weights, $\theta \in \mathbb{R}^{m \times d}$, where m is the number of units in the final hidden layer. 2) A Hebbian plastic component of the same cardinality as the slow weights, composed of the plasticity coefficient, α, and the Hebbian trace, Hebb. The α is a scaling parameter for adjusting the magnitude of the Hebb. Hebb accumulates the mean activations of the penultimate layer for each target label in the mini-batch $\{y_{1:B}\}$ of size B which are denoted by $\tilde{h} \in \mathbb{R}^{1 \times m}$ (refer to Algorithm 1). Given the activation of each neuron in h at the pre-synaptic connection i, the unnormalized log probabilities z at the post-synaptic connection j can be more formally computed using Eq. 2. Then, the softmax function is applied on z to obtain the desired logits $\hat{y}$ thus, $\hat{y} = \mathrm{softmax}(z)$. The η parameter in Eq. 3 is a "learning rate" that learns how quickly to acquire new experiences into the plastic component. The η parameter also acts as a decay term to prevent instability caused by a positive feedback loop in the Hebbian traces.

$$z_j = \sum_{i=1}^{m} (\underbrace{\theta_{i,j}}_{\text{slow}} + \underbrace{\alpha_{i,j}\mathrm{Hebb}_{i,j}}_{\text{plastic (fast)}})h_i \tag{2}$$

$$\mathrm{Hebb}_{i,j} := (1-\eta)\mathrm{Hebb}_{i,j} + \eta\tilde{h}_{i,j} \tag{3}$$

The network parameters $\alpha_{i,j}$, η and $\theta_{i,j}$ are optimized by gradient descent as the model is trained sequentially on different tasks in the continual learning setup. Hebb is initialized to zero only at the start of learning the first task T_1 and is automatically updated based on Algorithm 1 in the forward pass during training. Specifically, the Hebbian update for the active class c in $y_{1:B}$ is computed on line 6. This Hebbian update $\frac{1}{s}\sum_{b=1}^{B} h[y_b = c]$ is analogous to another formulaic description of the Hebbian learning update rule $w_{i,j} = \frac{1}{N}\sum_{k=1}^{N} a_i^k a_j^k$ (Hebb, 1949), where $w_{i,j}$ is the change in weight at connection i, j and a_i^k, a_j^k denote the activation levels of neurons i and j, respectively, for the k^{th} input. Therefore, in our model, $w = \tilde{h}$ the Hebbian weight update, $a_i = h$ the hidden activations of the last hidden layer, $a_j = y$ the active target class in $y_{1:B}$ and $N = s$ the number of inputs for the corresponding class in $y_{1:B}$ (see Algorithm 1). Across the model's lifetime, we only update Hebb during training and during test time, we use the most recent Hebbian traces to make predictions.

The plastic component learns rapidly and performs sparse parameter updates to quickly store memory traces for each recent experience without interference from other similar recent experiences. Furthermore, the hidden activations corresponding to the same active class are accumulated into one vector $\tilde{h}$, thus forming a compressed episodic memory in the Hebb to reflect individual episodic memory traces. This method improves learning of rare classes and speeds up binding of class labels to deep representations of the data.

Algorithm 1. Batch update Hebbian traces.

```
1:  Input: $h_{1:B}$ (hidden activations of penultimate layer),
           $y_{1:B}$ (target labels),
           $\mathrm{Hebb}$ (Hebbian trace)
2:  Output: $z_{1:B}$ (softmax pre-activations)
3:  for each target label $c \in \{y_{1:B}\}$ do
4:      $s \leftarrow \sum_{b=1}^{B}[y_b = c]$          /*Count total occurences of $c \in y$.*/
5:      if $s > 0$ then
6:          $\tilde{h} \leftarrow \frac{1}{s}\sum_{b=1}^{B} h[y_b = c]$   /*Update Hebb for active class c.*/
7:          $\mathrm{Hebb}_{:,c} \leftarrow (1-\eta)\mathrm{Hebb}_{:,c} + \eta\tilde{h}$
8:      end if
9:  end for
10: $z \leftarrow (\theta + \alpha\mathrm{Hebb})h$          /*Compute softmax pre-activations.*/
```

Updated Loss: Following the existing work for overcoming catastrophic forgetting such as EWC, Online EWC, SI and MAS (see Eq. 1), we regularize the loss $\mathcal{L}^n(\theta, \alpha, \eta)$ and update the synaptic importance parameters of the network in an online manner. We rewrite Eq. 1 to obtain Eq. 4 and show that the network parameters $\theta_{i,j}$ are the weights of the connections between pre- and post-synaptic activity, as seen in Eq. 2.

$$\begin{aligned}\tilde{\mathcal{L}}^n(\theta, \alpha, \eta) &= \mathcal{L}^n(\theta, \alpha, \eta) \\ &\quad + \lambda\sum_{i,j}\Omega_{i,j}(\theta_{i,j}^{n} - \theta_{i,j}^{n-1})^2\end{aligned} \tag{4}$$

We adapt these existing consolidation approaches to our model and only compute the synaptic importance parameters on the slow weights of the network. The plastic part of our model can alleviate catastrophic forgetting of learned classes by optimizing the plasticity of the synaptic connections.

4. Experiments

We tested our continual learning approach on the Permuted MNIST, Imbalanced Permuted MNIST and Split MNIST benchmarks. We evaluated the methods based on the average classification accuracy on all previously learned tasks. To establish a baseline for comparison of well-known synaptic consolidation methods, we trained neural networks with Online EWC, SI and MAS, respectively, on all tasks in a sequential manner. In the Permuted MNIST and Imbalanced Permuted benchmarks we trained a multi-layered perceptron (MLP) network on a sequence of 10 tasks using plain SGD. Detailed descriptions of the hyperparameters and training setups for all benchmarks can be found in Appendix A.

Permuted MNIST: In this benchmark, all of the MNIST pixels are permuted differently for each task with a fixed random permutation. Although the output domain is constant, the input distribution changes between tasks thus, there exists a concept drift. Figure 1 shows the average test accuracy as new tasks are learned. The network with DHP Softmax alone showed significant improvement in its ability to alleviate catastrophic forgetting across all tasks compared to the baseline finetuned vanilla MLP network we refer to as *Finetune* in Figure 1. Then we compared the performance with and without DHP Softmax using the synaptic consolidation methods. We find our DHP Softmax with synaptic consolidation maintains a higher test accuracy after T_{10} tasks than without DHP Softmax for all variants.

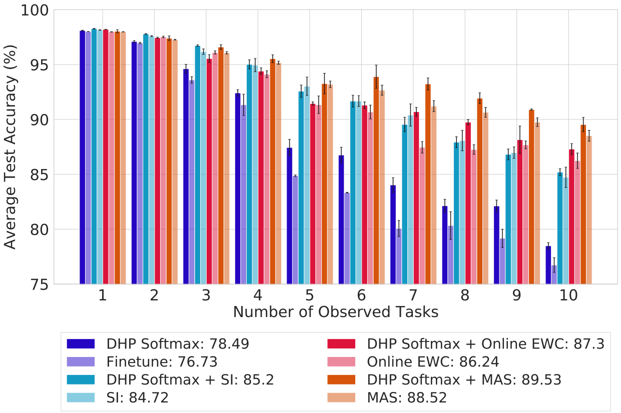

Figure 1. The average test accuracy on a sequence of Permuted MNIST tasks $T_{n=1:10}$. The average test accuracy after T_{10} tasks is given in the legend. Error bars correspond to SE on 10 trials.

Imbalanced Permuted MNIST: This benchmark is identical to the Permuted MNIST benchmark but, now each task is an imbalanced distribution. The statistics of the class distribution in each task are presented in Appendix A.2, Table 1. Figure 2 shows the average test accuracy as new tasks are learned. We see that DHP Softmax achieves 80.85% after learning 10 tasks, thus providing significant improvement over the standard neural network baseline of 76.4%. The significance of the compressed episodic memory mechanism in the Hebbian traces is more apparent in this benchmark because the plastic component allows rare classes that are encountered infrequently to be remembered for a longer period of time. We find that DHP Softmax with MAS achieves 88.8%; outperforming all other methods and across all tasks.

Split MNIST: A sequence of $T_{n=1:5}$ tasks are generated by splitting the original MNIST training dataset into binary classification problems (0/1, 2/3, 4/5, 6/7, 8/9), making the output spaces disjoint between tasks. Similar to Zenke et al. (2017), we trained a multi-headed MLP network on a sequence of 5 tasks. We compute the cross entropy loss at the softmax output layer only for the digits present in the current task, T_n. We observe that DHP Softmax provides a 4.7% improvement on test performance compared to a finetuned MLP network (Figure 3). Also, combining DHP Softmax with task-specific consolidation consistently improves performance across all tasks $T_{n=1:5}$.

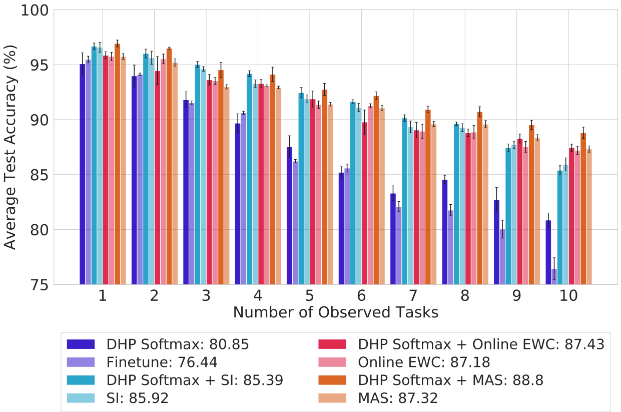

Figure 2. The average test accuracy on a sequence of *imbalanced* Permuted MNIST tasks $T_{n=1:10}$. The average test accuracy after T_{10} tasks is given in the legend. Error bars refer to SE on 10 trials.

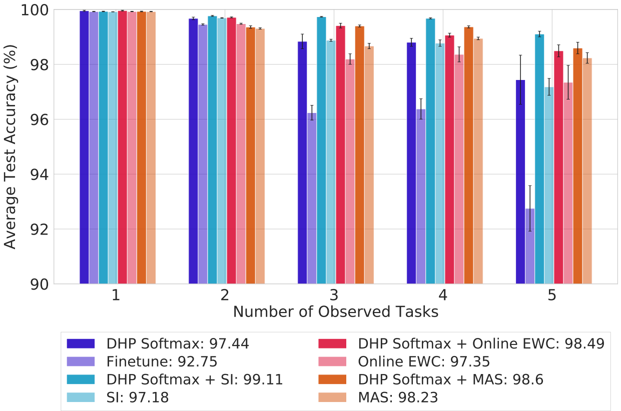

Figure 3. The average test accuracy on a sequence of 5 binary classification problems (0/1, 2/3, 4/5, 6/7, 8/9) from the original MNIST dataset. The average test accuracy after learning T_5 tasks is given in the legend. Error bars refer to the SE on 10 trials.

5. Discussion and conclusion

We have shown that the problem of catastrophic forgetting in continual learning environments can be alleviated by adding compressed episodic memory in the softmax layer through DHP. DHP Softmax alone showed noticeable improvement across all benchmarks when compared to a neural network with a traditional softmax layer. We demonstrated the flexibility of our model where, in addition to DHP Softmax, we can regularize the slow weights using EWC, SI or MAS to improve a model's ability to alleviate catastrophic forgetting. The approach where we combine DHP Softmax and MAS consistently leads to overall superior results compared to other baseline methods on several benchmarks. This gives a strong indication that Hebbian plasticity enables neural networks to learn continually and remember distant memories, thus reducing catastrophic forgetting when learning from sequential datasets in dynamic environments.

References

Abraham, W. C. and Robins, A. Memory retention – the synaptic stability versus plasticity dilemma. *Trends in Neurosciences*, 28(2):73–78, February 2005.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Ans, B., Rousset, S., French, R. M., and Musca, S. Self-refreshing memory in artificial neural networks: learning temporal sequences without catastrophic forgetting. *Connection Science*, 16(2):71–99, June 2004.

Atkinson, C., McCane, B., Szymanski, L., and Robins, A. V. Pseudo-recursal: Solving the catastrophic forgetting problem in deep neural networks. *CoRR*, abs/1802.03875, 2018.

Ba, J., Hinton, G. E., Mnih, V., Leibo, J. Z., and Ionescu, C. Using fast weights to attend to the recent past. In *Advances in Neural Information Processing Systems (NIPS) 29*, pp. 4331–4339. 2016.

Benna, M. K. and Fusi, S. Computational principles of synaptic memory consolidation. *Nature Neuroscience*, 19 (12):1697–1706, October 2016.

Carpenter, G. A. and Grossberg, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37(1):54–115, January 1987.

French, R. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, April 1999.

Hebb, D. O. *The organization of behavior; a neuropsychological theory.* Wiley, Oxford, England, 1949.

Hinton, G. E. and Plaut, D. C. Using fast weights to deblur old memories. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society*, pp. 177–186. Erlbaum, 1987.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)*, 114(13):3521–3526, March 2017.

Kumaran, D., Hassabis, D., and McClelland, J. L. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in Cognitive Sciences*, 20(7):512 – 534, 2016.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *IEEE Intelligent Signal Processing*, pp. 306–351. 2001.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, Dec 2018.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NIPS) 30*, pp. 6467–6476. 2017.

McClelland, J. L., McNaughton, B. L., and O'Reilly, R. C. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, July 1995.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24: 104–169, 1989.

Miconi, T. Learning to learn with backpropagation of hebbian plasticity. *CoRR*, abs/1609.02228, 2016.

Miconi, T., Stanley, K. O., and Clune, J. Differentiable plasticity: training plastic neural networks with backpropagation. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 3556–3565, 2018.

Miconi, T., Rawal, A., Clune, J., and Stanley, K. O. Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. In *International Conference on Learning Representations (ICLR)*, 2019.

Munkhdalai, T. and Trischler, A. Metalearning with hebbian fast weights. *CoRR*, abs/1807.05076, 2018.

Oja, E. Oja learning rule. *Scholarpedia*, 3(3):3612, 2008.

O'Reilly, R. C., Bhattacharyya, R., Howard, M. D., and Ketz, N. Complementary learning systems. *Cognitive Science*, 38(6):1229–1248, Aug 2014.

Paulsen, O. and Sejnowski, T. J. Natural patterns of activity and long-term synaptic plasticity. *Current Opinion in Neurobiology*, 10(2):172 – 180, 2000.

Rae, J. W., Dyer, C., Dayan, P., and Lillicrap, T. P. Fast parametric learning with activation memorization. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 4225–4234, 2018.

Ring, M. B. *Continual Learning in Reinforcement Environments*. PhD thesis, Austin, TX, USA, 1994. UMI Order No. GAX95-06083.

Robins, A. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, June 1995.

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. Progress & compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 4535–4544, 2018.

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems (NIPS) 30*, pp. 2990–2999. 2017.

Song, S., Miller, K. D., and Abbott, L. F. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9):919–926, September 2000.

Thrun, S. and Mitchell, T. M. Lifelong robot learning. *Robotics and Autonomous Systems*, 15(1):25 – 46, 1995. The Biology and Technology of Intelligent Autonomous Agents.

Wu, C., Herranz, L., Liu, X., Wang, Y., van de Weijer, J., and Raducanu, B. Memory replay gans: Learning to generate new categories without forgetting. In *Advances in Neural Information Processing Systems (NeurIPS) 31*, pp. 5962–5972. 2018.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 3987–3995, 2017.

A. Appendix

The model used for the Permuted MNIST and Imbalanced Permuted MNIST benchmarks is a multilayered perceptron (MLP) network with two hidden layers consisting of 400 units each with ReLU nonlinearities, and a cross-entropy loss $\mathcal{L}(\theta)$. We train the network on a sequence of 10 tasks $T_{n=1:10}$ with mini-batches of size 64 and optimized using plain SGD with a fixed learning rate of 0.01. We train for atleast 10 epochs and perform early-stopping once the validation error does not improve for 5 epochs. If the validation error increases for more than 5 epochs, then we terminated the training on the task T_n, reset the network weights and Hebbian traces to the values that had the lowest validation error, and proceeded to the next task.

For all of the benchmarks we tested on, the η of the plastic component was set to be a small value of 0.001 and we want to emphasize that we spent little to no efforts on tuning this parameter. Also, when training the first task $T_{n=1}$, the synaptic importance parameter, $\Omega_{i,j}$ in Eq. 4, was set to 0 for all of the task-specific consolidation methods that we tested on except for SI. This is because SI is the only method we evaluated that estimates $\Omega_{i,j}$ while training, whereas Online EWC and MAS compute $\Omega_{i,j}$ after learning a task.

A.1. Permuted MNIST

For the Permuted MNIST experiments shown in Figure 1, the regularization hyperparameter λ for each of the task-specific consolidation methods is $\lambda = 100$ for Online EWC (Schwarz et al., 2018), $\lambda = 0.1$ for SI (Zenke et al., 2017) and $\lambda = 0.1$ for MAS (Aljundi et al., 2018). In SI, the damping parameter, ξ, was set to 0.1. To find the best hyperparameter combination for each of these synaptic consolidation methods, we performed a grid search using a task sequence determined by a single seed. The hyperparameters of the consolidation methods (i.e. Online EWC, SI and MAS) remain the same with and without DHP Softmax, and the plastic components are not regularized.

A.2. Imbalanced Permuted MNIST

For each task in the Imbalanced Permuted MNIST problem, we artificially removed training samples from each class in the original MNIST dataset (LeCun et al., 2001) based on some random probability. The distribution of classes in each dataset corresponding to tasks $T_{n=1:10}$ is given in Table 1.

Table 1. Distribution of classes in each imbalanced dataset for the respective tasks $T_{n=1:10}$.

	TASKS									
CLASSES	1	2	3	4	5	6	7	8	9	10
0	4459	3780	1847	3820	5867	122	1013	4608	908	3933
1	1872	3637	1316	6592	1934	1774	5533	2569	831	886
2	2391	4125	2434	4966	5245	4593	4834	4432	3207	3555
3	4433	1907	1682	278	3027	2315	5761	3293	2545	3749
4	186	2728	2002	151	1435	5829	1284	3910	4593	927
5	4292	2472	2924	1369	4094	4858	2265	3289	1134	1413
6	2339	3403	4771	5569	1414	2851	2921	4074	336	3993
7	4717	3090	4800	2574	4086	1065	3520	4705	5400	3650
8	3295	5493	76	4184	2034	4672	682	196	2409	1709
9	2625	3880	4735	1647	2645	3921	901	4546	4649	2045
TOTAL	30609	34515	26587	31120	31781	32000	28714	35622	26012	25860

For the Imbalanced Permuted MNIST experiments shown in Figure 2, the regularization hyperparameter λ for each of the task-specific consolidation methods is $\lambda = 400$ for Online EWC (Schwarz et al., 2018), $\lambda = 1.0$ for SI (Zenke et al., 2017) and $\lambda = 0.1$ for MAS (Aljundi et al., 2018). In SI, the damping parameter, ξ, was set to 0.1. Similar to the Permuted MNIST benchmark, to find the best hyperparameter combination for each of these synaptic consolidation methods, we performed a grid search using a task sequence determined by a single seed. Across all experiments, we maintained the the same random probabilities detemined by a single seed to artificially remove training samples from each class. The hyperparameters of the synaptic consolidation methods (i.e. Online EWC, SI and MAS) remain the same with and without DHP Softmax, and the plastic components are not regularized.

A.3. Split MNIST

We split the original MNIST dataset (LeCun et al., 2001) into a sequence of 5 binary classification tasks: $T_1 = \{0/1\}$, $T_2 = \{2/3\}$, $T_3 = \{4/5\}$, $T_4 = \{6/7\}$ and $T_5 = \{8/9\}$. Similar to network used by Zenke et al. (2017), we use a MLP network with two hidden layers of 256 ReLU nonlinearities each, and a cross-entropy loss. A multi-headed approach was used to avoid interference between digits at the softmax output layer due to changes in the label distribution. We compute the cross-entropy loss, $L(\theta)$, at the softmax output layer for the digits present in the current task, T_n. We train the network on a sequence of 5 tasks $T_{n=1:5}$ with mini-batches of size 64 and optimized using plain SGD with a fixed learning rate of 0.01 for 10 epochs.

For the Split MNIST experiments shown in Figure 3, the regularization hyperparameter λ for each of the task-specific consolidation methods is $\lambda = 400$ for Online EWC (Schwarz et al., 2018), $\lambda = 1.0$ for SI (Zenke et al., 2017) and $\lambda = 1.5$ for MAS (Aljundi et al., 2018). In SI, the damping parameter, ξ, was set to 0.001. To find the best hyperparameter combination for each of these synaptic consolidation methods, we performed a grid search using the 5 task binary classification sequence (0/1, 2/3, 4/5, 6/7, 8/9). The hyperparameters of the consolidation methods (i.e. Online EWC, SI and MAS) remain the same with and without DHP Softmax, and the plastic components are not regularized.