Multilayer Neuromodulated Architectures for Memory-Constrained Online Continual Learning

Sandeep Madireddy [1] Angel Yanguas-Gil [2] Prasanna Balaprakash [1]

Abstract

We focus on the problem of how to achieve online continual learning under memory-constrained conditions where the input data may not be known *a priori*. These constraints are relevant in edge computing scenarios. We have developed an architecture where input processing over data streams and online learning are integrated in a single recurrent network architecture. This allows us to cast metalearning optimization as a mixed-integer optimization problem, where different synaptic plasticity algorithms and feature extraction layers can be swapped out and their hyperparameters are optimized to identify optimal architectures for different sets of tasks. We utilize a Bayesian optimization method to search over a design space that spans multiple learning algorithms, their specific hyperparameters, and feature extraction layers. We demonstrate our approach for online non-incremental and class-incremental learning tasks. Our optimization algorithm finds configurations that achieve superior continual learning performance on Split-MNIST and Permuted-MNIST data as compared with other memory-constrained learning approaches, and it matches that of the state-of-the-art memory replay-based approaches without explicit data storage and replay. Our approach allows us to explore the transferability of optimal learning conditions to tasks and datasets that have not been previously seen. We demonstrate that the accuracy of our transfer metalearning across datasets can be largely explained through a transfer coefficient that can be based on metrics of dimensionality and distance between datasets.

1 Introduction

The ability to learn and generalize from a few examples is present on a wide range of species, from the fruit fly to humans. It is therefore not surprising that biological systems have served as inspiration for the exploration of continual or incremental learning. How to achieve the right balance between plasticity and consolidation, and therefore how to avoid catastrophic forgetting, has long been one of the central areas of research in the area of continual learning. However, the design of systems intended for edge computing brings up an additional set of constraints, such as how to design systems that have to learn inputs that are not known during the design phase, how to achieve high accuracy when the system has no external way of replaying data (i.e., stream processing), or what the trade-off is between performance and size.

These resource constraints may cause performance penalties with respect to online and continual learning strategies that are not bounded by memory and that keep a detailed record of past performance and data. On the other hand, these constraints are also present in biological systems, and yet they excel in many continual learning tasks. Neuro-inspired approaches can therefore provide useful insights that complement other machine learning approaches.

One of the salient features of the central nervous systems is its heterogeneity in terms of architecture, types of neurons, and neurochemistry, including where, when, and how learning takes place within the brain. If we focus on synaptic plasticity mechanisms, we see a wide diversity, not only in the hyperparameters of a specific mechanism or in where learning takes place, but also in the nature of the synaptic plasticity mechanism itself. For instance, while Hebbian rules and spike timing-dependent plasticity have attracted the bulk of the attention in the machine learning and neuromorphic computing communities, anti-Hebbian and non-Hebbian rules abound (Dayan, 2001). For instance, in some key learning centers researchers have showed that synaptic plasticity is independent of postsynaptic activity (Hige et al., 2015). Moreover, variations in the local learning rules have been found even in areas that are otherwise morphologically equivalent but that are assigned to different tasks. Clearly,

[1] Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, Illinois [2] Applied Materials Division, Argonne National Laboratory, Lemont, Illinois. Correspondence to: Sandeep Madireddy <smadireddy@anl.gov>.

4th Lifelong Machine Learning Workshop at 37^{th} International Conference on Machine Learning, 2020.

if we want to develop robust continual learning approaches, we need to pay attention to this diversity and develop a better understanding of the interplay between heterogeneity and continual learning across very different tasks, which is often overlooked in continual learning across diverse tasks.

To this end, we make the following contributions.

- We propose a neuromorphic architecture that consists of multiple layers for feature extraction and neuromodulated supervised learning. It incorporates diverse synaptic plasticity mechanisms, and hence is capable of online continual learning without explicit data storage and replay.

- We cast the problem of metalearning task-specific architecture choice as a mixed-integer black-box optimization problem and develop a technique based on transfer coefficients to explain the accuracy differences when the metalearned configurations are transferred across datasets and tasks.

- We evaluate our approach in an online non-incremental learning and class-incremental learning scenarios and demonstrate that the accuracy values are on par with the shallow networks in the former scenario and outperform other memory-free continual learning approaches while obtaining accuracies similar to those of the memory-replay-based approaches without the memory.

2 Multilayer Neuromodulated Architecture

We consider multilayer, recurrent architectures integrating both processing and learning into the networks as shown in Fig. 1.

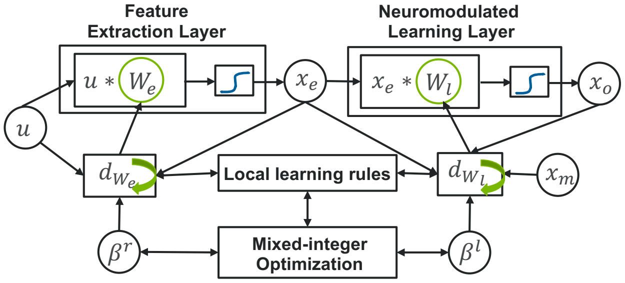


Figure 1: Multilayer neuromodulated architecture that consists of feature extraction and neuromodulated learning layers that incorporate synaptic plasticity mechanisms through local learning rules.

2.1 Processing Component

The processing component comprises a feed-forward neural network. It has two layers:

Feature extraction layers: It is based on sparse projection (SPARSE) of the N-dimensional input into a much larger M-dimensional space. Each neuron receives K inputs, with $K \ll N$. In addition, dynamic thresholding followed by rectification ensures that only the most salient features are included in the representation:

$$\mathbf{v} = \mathrm{ReLU}\left(\mathrm{W_e}\mathbf{u} - \mu - \beta_r^1 \sigma\right), \tag{1}$$

where $\mathrm{W_e}$ is the sparse projection matrix, μ and σ are the mean and standard deviation of the matrix multiplication product $\mathrm{W_e}\mathbf{u}$, and β_r^1 is a constant controlling the cutoff. This approach highlights correlations between different input channels and represents the broadest possible prior for feature extraction, since it does not assume any spatial dependence or correlations between inputs. This layer is similar to that used in previous models inspired in the cerebellum and the mushroom body, two of the biological instances of this architecture (Litwin-Kumar et al., 2017; Yanguas-Gil, 2019). A key distinction is that $\mathrm{W_e}$, K, and β_r^1 can all be adjusted through our optimization algorithm.

Neuromodulated learning layer: This comprises one or more layers where supervised learning occurs. Synaptic weights are treated as first-class network elements that can be updated in real time by using a series of local learning rules codified in the learning component.

2.2 Learning Component

It comprises the recurrent version of the network, implementing local learning rules. These local learning rules are defined as functions:

$$\mathrm{W}_l(t+1) = F\left(\mathrm{W}_l(t), \mathbf{x}_e, \mathbf{x}_o, \mathbf{x}_m; \beta^l\right), \tag{2}$$

where W represents the synaptic weights; $\mathbf{x}_e$ and $\mathbf{x}_o$ are the presynaptic and postsynaptic activities, respectively; $\mathbf{x}_m$ represents a vector of modulatory signals controlling learning in real time; and β^l represents a discrete number of hyperparameters that are specific to a given learning rule. By using a consistent interface, we can swap out learning rules without having to further modify our architecture. This allows us to treat metalearning as a combined architecture and hyperparameter optimization process.

The architecture receives real-time feedback on its performance while learning (i.e., the label). The modulatory component transforms this feedback into the modulatory signals that are fed into the local learning rules. Here, we use the following synaptic plasticity rules.

- *Generalized Hebbian model (GEN):* the generalized Hebbian model is a modulated version of the covariance rule commonly used in neuroscience. The synaptic weight evolution is given by

$$\tau \frac{dW_l}{dt} = \mathbf{x}_m(\mathbf{x}_e - \beta_1^l)(\mathbf{x}_o - \beta_2^l). \tag{3}$$

It has long been known that this rule is unstable and that clamping of a regularization mechanism needs to be included in order to keep the synaptic weights bounded. This rule is at the core of some of the neuromodulation-based approaches recently developed (Miconi et al., 2018).

- *Oja rule (OJA):* a modification of the basic Hebb's rule providing a normalization mechanisms through a first-order loss term (Oja, 1982):

$$\tau \frac{dW_l}{dt} = \mathbf{x}_m(\mathbf{x}_e \mathbf{x}_o - \beta_1^l x_o^2 W_l). \tag{4}$$

- *MSE, non-Hebbian rule:* It is based on recent experimental results on synaptic plasticity mechanisms in the mushroom body (Hige et al., 2015). The key assumptions are that learning is independent of postsynaptic activity and that instead postsynaptic activity modulates regulates synaptic plasticity:

$$\tau \frac{dW_l}{dt} = (\mathbf{x}_m - \mathbf{x}_o)\mathbf{x}_e. \tag{5}$$

This rule is consistent with the MSE cost function used in stochastic gradient descent methods (Yanguas-Gil et al., 2019).

These are just a few of the many nonequivalent possible formulations of synaptic plasticity rules (Madireddy et al., 2019). We have treated these models as categorical variables that can be swapped during the metalearning optimization process. Moreover, we have added two conditions that allow us to externally tune the learning rate as a function of time and to provide symmetric or positive clamping of the synaptic weights in order to prevent instabilities in the algorithms.

2.3 Mixed-Integer Optimization Framework

The parameter space in the proposed multilayer neuromodulated architecture is composed of categorical variables (e.g., the selection of the local learning rule), integer parameters (e.g., the dimension of hidden layer), and continuous parameters in each of the learning rules. We adopt a parallel asynchronous-model-based search approach (AMBS) (Balaprakash et al., 2018) to find the high-performing parameter configuration in this mixed (categorical, continuous, integer) search space. Our AMBS approach consists of sampling a number of parameter configurations and progressively fitting a surrogate model over the parameter configurations' accuracy metric space. This surrogate model is asynchronously updated as new configurations are being evaluated by the parallel processes, which are then used to obtain configurations that will be evaluated in the next iteration.

Crucial to the optimization approach is the choice of the surrogate model, since this model generates the configuration to evaluate in the mixed search space. The AMBS adopts *random forest* approach to build efficient regression models on this search space. The random forest is an ensemble learning approach that builds multiple decision trees and uses bootstrap aggregation (or bagging) to combine them to produce a model with better predictive accuracy and lower variance. Another key choice for the AMBS approach is the acquisition function, which encapsulates criteria to choose the most promising configurations to evaluate next. Hence, the acquisition function is key to maintaining the exploration-exploitation balance during the search. The AMBS adopts the *lower confidence bound* (LCB) acquisition function. We set the kappa to a large value, which increases exploration. We do so to accommodate the high variability we observed in the accuracy metrics within and across the local learning rules that led them to local minimum when a smaller value of kappa was used.

3 Online and Continual Learning Experiments

We focus on a specific type of experiment in which the system is subject to a stream of data and labels during a predetermined number of epochs. This constitutes a single episode. Through the synaptic plasticity mechanisms implemented in the network, the weights evolve during the episode. At the end of the episode, the system is evaluated against the testing dataset to validate its accuracy. By concatenating multiple episodes involving different tasks and datasets, we can create a curriculum to evaluate the system's ability to carry out continual learning.

The resulting accuracy is the metric that the optimization framework uses to carry out the exploration of the architecture and hyperparameter space and find the optimal configurations. As part of the optimization process, multiple episodes are run for systems with different architectures and specific hyperparameters. In all these cases, the system starts from identical starting conditions, so that no knowledge is transferred between episodes that do not belong to the same curriculum.

We consider two different learning modalities to evaluate the proposed approach:
(a) Non-Incremental Learning: There is a single episode in the curriculum, which has access to all the data and hence performs a single task (e.g., multiclass classification);
(b) Class-Incremental Learning: The synaptic weights are updated incrementally, where a specific task (e.g., two-

class classification) is performed in each episode using data streams from distinct (clearly separated) classes. The curriculum involves a concatenation of multiple episodes such that the final accuracy measures the model's ability to predict on test data from all the classes, spanning across episodes.

By changing tasks, datasets, and learning modalities, we build a collection of high-performing configurations for each type of experiment. We use these configurations to explore the transferability of optimal learning conditions to other datasets and to different types of tasks.

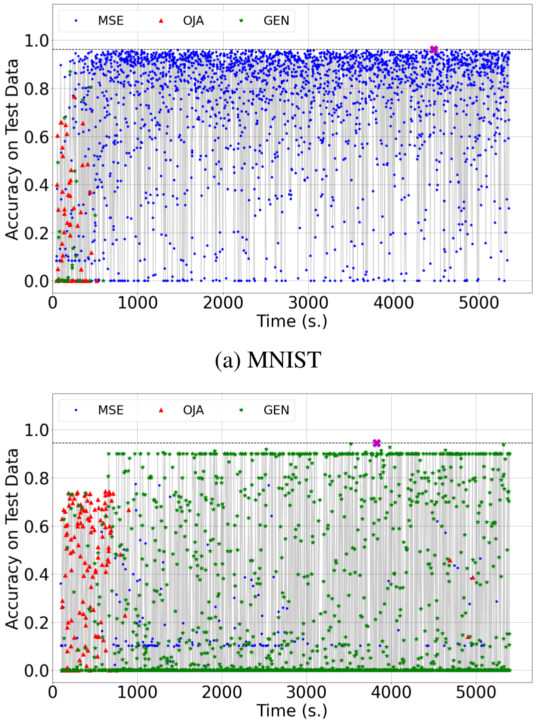
4 Results and Discussion

We first describe the results obtained in the non-incremental learning scenario, followed by incremental learning scenario. For both scenarios, we demonstrate that the accuracies we obtain are on par with (or outperformed) other state-of-the-art approaches without explicit memory consideration. Finally, we perform experiments exploring the transferability of optimal learning conditions to tasks and datasets and find that it is strongly correlated to similarity between the datasets.

4.1 Non-Incremental Learning

In this experiment, we demonstrate the learning capability of our multilayer neuromodulated learning framework on single episode curriculum learning task, specifically multi-class classification. We consider MNIST, Fashion MNIST (Xiao et al., 2017b) (F-MNIST), and Extended MNIST (Cohen et al., 2017) (E-MNIST) datasets because of the homogeneity in image and class sizes and the existence of benchmarks in continual learning. For each dataset, we jointly optimize over the local learning rules and their parameters to find the optimal configuration.

We employed SPARSE as the feature representation layer and three supervised learning rules (MSE, GEN, OJA) for label prediction. While the number of parameters in each of the learning rule differs, the search space is defined to exploit the common parameters, whose dimensionality is defined by the rule with maximum parameters. In addition, only the subset of parameters present in the local-learning rule are active at search time. For example, the parameter β_l^3 defines the weight clamping and is common to the three rules; the GEN rule has all the parameters ($\beta_l^1, \beta_l^2, \beta_l^3$) active; whereas MSE has only β_l^3 active, so the other parameters (β_l^1, β_l^2) are set to *None* while exploring the search space. For each configuration evaluated during the optimization, the model is run for 0.5 epochs. The best accuracy obtained and the corresponding optimal parameters for all the datasets are shown in Table 1. The search trajectory showing the learning rule and test accuracy of the configurations evaluated as a function of time is shown Fig. 2 for MNIST and F-MNIST. In both the cases, the initial configurations range across the different learning rules, but the algorithm quickly finds the potential learning rule and evaluates more configurations from it. The corresponding accuracies for each of the three datasets obtaining by using the optimal configurations, but run of 4 epochs is shown in Table 2. These accuracies are on par with (or outperform) other shallow network architectures (Yanguas-Gil et al., 2019; Xiao et al., 2017a; Cohen et al., 2017).



(a) MNIST

(b) F-MNIST

Figure 2: Search trajectory obtained for MNIST and F-MNIST datasets from the mixed-integer black-box optimization. The configurations are colored by their evaluated learning rule, with blue, red and green corresponding to MSE, OJA, and GEN respectively.

4.2 Incremental Learning

In this experiment, we demonstrate the learning capability of our approach in the class-incremental learning scenario, where multiple episodes are present per learning curriculum. We evaluate our method on the Split-MNIST (Farquhar & Gal, 2018) and Permuted-MNIST (Goodfellow et al., 2013) incremental learning benchmarks that have been extensively adopted in the literature (Shin et al., 2017; Zenke et al., 2017; Nguyen et al., 2017). The Split-MNIST data is prepared by splitting the original MNIST dataset (both training and testing splits) consisting of ten digits into five two-class classification tasks. This defines an incremental learning sce-

Table 1: Optimal configurations obtained for MNIST, F-MNIST, and E-MNIST datasets using the mixed-integer black-box optimization approach, where each configuration was evaluated after 0.5 epochs of training.

Dataset	Acc.	Srule	N_H^F	β_1^F	β_2^F	β_3^F	γ^F	α^F	β_1^M	β_2^M	β_3^M	γ^M	α^F
MNIST	96.20	MSE	9000	0.022	0.326	1.658	0.968	0.225	–	–	0.585	0.013	0.260
F-MNIST	94.38	GEN	7000	0.957	0.003	6.025	0.332	0.578	0.083	0.103	0.400	0.810	0.077
E-MNIST	96.41	MSE	9000	0.027	0.306	1.918	0.984	0.678	–	–	0.531	0.189	0.350

Table 2: Classification accuracy in the non-incremental learning scenario for MNIST, F-MNIST, and E-MNIST datasets after four epochs using the optimal configurations learned through the optimization framework.

MNIST	F-MNIST	E-MNIST
97.45	94.32	97.34

nario in which the model sees these five tasks incrementally one after the other. The Permuted-MNIST is also prepared from the original MNIST data by permuting the pixels in the images. A unique permutation is applied to all the images in order to generate a new ten-class classification task. The number of distinct permutations applied represents the length of the task sequence. For instance, we adopted a ten-permuted MNIST, which produces ten (ten-class classification) tasks that are sequentially seen by the model. The first task is the original MNIST data, and the subsequent nine tasks are obtained by permuting the original data.

Following (Hsu et al., 2018), for Split-MNIST, a simple two-layer multilayered perceptron (MLP) neural network with 400 neurons per layer is adopted to evaluate the baseline accuracy, as well as the accuracy with other incremental learning algorithms. Similarly, for Permuted-MNIST, we use two-layer MLP with 1,000 neurons per layer to evaluate the accuracy of the baseline and other incremental learning algorithms. See (Hsu et al., 2018) for more details on the hyperparameters used for both these models. The first baseline is a naive approach in which the MLP network (with Adagrad optimizer) is trained by progressively updating the parameters through backpropagation as new tasks are observed. The second baseline is a naive rehearsal (experience replay) approach that stores a fraction of data from previous tasks randomly and uses that while training incrementally. Among the popular incremental learning approaches, we choose representative ones from regularization-based (Online EWC (Schwarz et al., 2018), SI (Zenke et al., 2017), MAS (Aljundi et al., 2018)), and memory-based methods (GEM (Lopez-Paz & Ranzato, 2017), DGR (Shin et al., 2017), Rtf (van de Ven & Tolias, 2018)). We also compare the non-incremental learning scenario with the corresponding baseline models for both datasets as well as our multilayer neuromodulated learning approach. The results of the incremental learning experiments are shown in Table 3.

For the non-incremental learning case, where the data from all tasks are provided at the same time, the baseline MLP model gives an accuracy of 97.53% and 97.95% for the Split-MNIST and Permuted-MNIST datasets, respectively, after 4 epochs of training for the former and 10 epochs for the latter, following (Hsu et al., 2018). We use the best-performing parameter configuration for MNIST data obtained for the non-incremental learning (Table 1) to evaluate the performance of our architecture in these two datasets. This transfer of parameter configuration from the non-incremental to the incremental learning scenario is further discussed in the next section. With this parameter configuration, we found that the accuracy of our model is 97.45 on Split-MNIST trained for 4 epochs and 96.69% on the Permuted-MNIST trained for just 1 epoch. These values are very close to the MLP model accuracy—97.53 on Split-MNIST trained for 4 epochs and 97.95% on the Permuted-MNIST trained for 10 epochs .

For the class-incremental learning on Split-MNIST and the Permuted-MNIST, our model (using MNIST parameter configuration from Table 1) significantly outperforms both the baseline and other non-memory-based incremental learning approaches considered. In addition, we obtain accuracy comparable to the state-of-the-art memory-based models for Split-MNIST data and outperform all the other incremental learning algorithms on the Permuted-MNIST data with only three epochs (as compared with ten epochs for all the other algorithms). The naive rehearsal approach outperforms all the other approaches, but it comes with the additional memory overhead.

4.3 Transferability Study

Identification of the best configuration for each group of classes (a dataset) is based on the assumption that all the data is available at the beginning of the training procedure. However, for many online learning scenarios, both incremental and non-incremental, this configuration might not be known *a priori*. This raises the question of transfer metalearning: how to effectively optimize a learning algorithm to learn unknown tasks and data. To address this problem, we empirically study the effect of transferring optimal configurations tasks for the incremental learning scenario, and

Table 3: Classification accuracy comparison for the class-incremental learning experiments on Split-MNIST and Permuted-MNIST datasets.

	Method	Split-MNIST	Permuted-MNIST
Non-incremental	MLP	97.53	97.95
	SPA+MSE	97.45	96.69 (1 epoch)
Baseline	Adagrad	19.75	79.50
	Naive rehearsal-C (*)	94.35	97.15
Incremental Learning (*) → Memory-based	Online EWC	19.71	86.57
	SI	20.88	79.36
	MAS	19.98	73.82
	GEM (*)	92.20	96.72
	DGR (*)	91.24	92.19
	Rtf (*)	92.56	96.23
	SPA+MSE	92.76	96.62 (1 epoch)

across datasets for the non-incremental learning case.

Across tasks: The best configuration obtained through the joint optimization on each dataset separately (Table 1) is used to evaluate the incremental learning accuracy with Split-MNIST and Permuted-MNIST datasets. The results using the configurations corresponding to MNIST, F-MNIST, and E-MNIST are shown in Table 4. We observe that the configuration learned on the MNIST dataset is readily transferable to incremental learning on Split-MNIST and the Permuted-MNIST data, as seen in the preceding section. The configuration learned with the E-MNIST data leads to an accuracy difference of only 0.71% and 0.50%, respectively, on these incremental learning datasets. The F-MNIST configuration, on the other hand, leads to an accuracy drop of 0.98% and 4.14%, respectively on them. This observation suggests that the transfer learning configuration (both the local learning rule and its corresponding parameters) to the incremental learning experiments needs to be done carefully, which otherwise would lead to suboptimal accuracy.

Across datasets: We seek to understand and characterize the dependence of dataset similarity (defined as a distance metric) and the transferability of optimal configurations learned on a dataset to other (different) datasets. To this end, we choose the best configuration obtained through the joint optimization on each dataset (Table 1) and use that for transfer learning and evaluating the accuracy for the remaining datasets. The results for transfer learning across MNIST, F-MNIST, and E-MNIST are summarized in Table 4, where a column represents the accuracy on a particular dataset obtained by transfer learning through the optimal configurations learned on standalone MNIST, F-MNIST, and E-MNIST datasets. The diagonal elements are the same as the accuracy values obtained previously for four epoch runs with the same data-optimal configuration combination (Table 2). We find that the configurations learned on MNIST and E-MNIST are readily exchangeable without significant loss in accuracy. These configurations, however, do not transfer well to the F-MNIST dataset, as seen by the accuracy decrease of 11.45% and 11.12%, respectively, with MNIST and E-MNIST. On the other hand, the F-MNIST configurations also lead to a decrease in accuracy of 5.53% and 3.75%, respectively, when transferred to MNIST and E-MNIST data.

Distance metrics as a measure of transferability: To rationalize the results shown in Table 4, we have explored the correlation between the drop in performance between datasets and their effective dimensionality and the distance between datasets. We focus on a definition of dimensionality derived from the covariance matrix of the complete dataset. The eigenvalues of the covariance matrix yields the principal components of the dataset. We consider as a metric of dimensionality the sum of squares of the eigenvalues (λ), so that

$$d = (\sum_i \lambda_i)^2 / \sum_i \lambda_i^2. \tag{6}$$

This metric has been used in the past to characterize sparse representations (Litwin-Kumar et al., 2017). The interpretation of this metric is that when the variance is equally distributed across all dimensions, Eq. 6 yields a dimension $d = N$, where N is the number of dimensions in the space (784 in this case). If all the variance is concentrated in a single eigenvalue, then $d = 1$.

To quantify the distance and separability between categories in the dataset, we have used the cosine distance metric. In Table 6 we have applied the traditional definition of distance between two clusters and calculated the minimum distance between any two categories of two datasets. We also show the maximum separation for comparison. When these metrics are used within a given dataset, we obtain the minimum and maximum distance between two categories of

Table 4: Accuracy due to transfer metalearning in class-incremental and non-incremental learning experiments.

	Non-Incremental Learning			Class-Incremental Learning	
Config (↓), Dataset (→)	MNIST	F-MNIST	E-MNIST	SplitMNIST	P-MNIST
MNIST	97.35	82.87	97.69	92.76	96.62
F-MNIST	91.82	94.32	93.59	91.78	92.48
E-MNIST	96.93	83.20	97.34	93.47	96.12

Table 5: PCA metric of dimensionality for each of the datasets (Eq. 6)

Data	MNIST	F-MNIST	E-MNIST
Eigen-Dim	30.69	7.91	27.09

Table 6: Minimum and maximum cosine distances between centroids of the categories of different datasets

Data	MNIST	F-MNIST	E-MNIST
MNIST	(0.073,0.55)	(0.17,0.54)	(0.044,0.59)
F-MNIST		(0.012,0.56)	(0.13,0.57)
E-MNIST			(0.098,0.52)

the same dataset, which could be interpreted as a measure of separability.

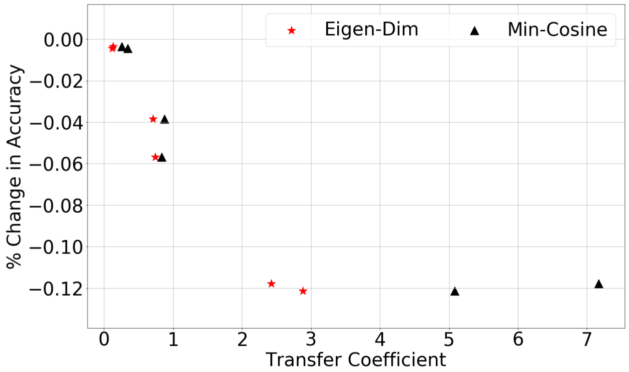


Figure 3: Drop in classification accuracy during transfer metalearning as a function of a transfer coefficient obtained for PCA metric and minimum cosine distance metric.

Figure 3 shows the drop in classification accuracy during transfer metalearning as a function of a *transfer coefficient* obtained from two different metrics: the relative difference in the eigenvalue dimension given by Eq. 6 and the minimum cosine distance between the two datasets. Both values are normalized as

$$D = |M_1 - M_2|/M_1, \qquad (7)$$

where M_1 is the metric of the dataset selected to run the experiment and M_2 is the metric of the dataset whose optimal configuration is used. Note that this transfer coefficient is not symmetric because of the different normalization in Eq. 7, as should be expected since transfer metalearning is directional. The results clearly show that as the distance between the datasets increases, transfer learning the configurations across them leads to a decrease in accuracy.

5 Related Work

Incremental learning, also referred to as continual or lifelong learning (Thrun & Pratt, 2012), describes a learning modality in which a model seeks to learn from data and tasks that are sequentially presented to it. Several incremental learning approaches have been presented in the literature, which can loosely be categorized into three classes: (1) novel neural architectures or customization of the common ones; (2) regularization strategies that impose constraints to boost knowledge retainment; and (3) metalearning, which uses a series of tasks to learn a common parameter configuration that is easily adaptable for new tasks. Algorithms in the first category include bio-inspired dual-memory architecture (Parisi et al., 2019); progressive neural networks (Rusu et al., 2016) that explicitly support information transfer across sequences of tasks through network expansion; and deep generative replay (Shin et al., 2017), which proposed a cooperative dual model architecture framework, inspired by hippocampus, that retains past knowledge by the concurrent replay of generated pseudodata. The second category consists of algorithms such as elastic weight consolidation (EWC) (Kirkpatrick et al., 2017) that computes synaptic importance using a Fisher importance matrix-based regularization; synaptic intelligence (SI) (Zenke et al., 2017), whose regularization penalty is similar to EWC but is computed online at per-synapse level; and learning without forgetting (Li & Hoiem, 2017), which applies a distillation loss on the attention-enabled deep networks seeking to minimize task overlap. The third category consists of algorithms such as online metalearning (Javed & White, 2019), neuromodulated metalearning algorithm (Beaulieu et al., 2020), and incremental task-agnostic metalearning (Rajasegaran et al., 2020) that show great promise but are not particularly suited for the memory-constrained continual learning scenarios that could be relevant in edge computing. These algorithms are characterized by large network sizes and memory buffer requirements and have a few restricting assumptions on the structure of the data; hence we do not include a comparison with this class of approaches. The multilayer neuromodulated architecture approach we proposed falls under the first class of continual learning methods discussed earlier.

6 Conclusions

We developed a multilayer, recurrent neuromorphic architecture capable of online continual learning in a memory-constrained setting, where large models and data storage/replay is limited. The proposed architecture consists of a processing component and a learning component, where learning can be viewed as a dynamic process that alters the architecture itself via recurrent interactions as it processes and assigns valence to certain inputs and create associations over time. Our approach parameterizes multiple synaptic plasticity mechanisms as layers of this network sharing a common interface. This allows us to cast the optimization of the architecture's learning capabilities as an optimization problem and to employ a Bayesian optimization-based search to find optimal task-specific configurations in the mixed (categorical, continuous) integer space that spans over the choice of the learning algorithms, their specific hyperparameters, and feature extraction layers.

We demonstrate our approach using two different learning scenarios. The first is a non-incremental learning scenario, where the learning curriculum consists of a episode that has access to all tasks, and the model is updated online as the data streams in. The second is a class-incremental learning scenario, where data from each task is presented sequentially such that each episode in the learning curriculum consists of a unique task. The model is then updated online in each episode and continually with a sequence of these episodes. In the non-incremental learning case, our algorithm identified configurations capable of obtaining online learning accuracies of 97.45 for MNIST, 94.32 for Fashion MNIST, and 97.34 for Extended MNIST, which were on par with (or outperformed) the performance of static shallow neural networks run for four epochs. The optimal configurations in our architecture were obtained by imposing a limit of 0.5 epochs to learn each dataset, which was cheap computationally and led to good configurations. In the class-incremental learning case, we obtained an accuracy of 92.76 and 96.62 on the Split-MNIST and Permuted-MNIST data, which consisted of five and ten sequential tasks, respectively. Hence, our approach clearly outperformed the memory-free approaches, whose accuracies obtained a maximum of 20.88 and 85.57, respectively. The memory replay-based continual learning approaches gave a maximum accuracy of 92.56 with Rtf and 96.72 with GEM using four and ten epoch of training, respectively, on the two datasets, whereas our approach produced an accuracy of 92.76 with four epochs on Split-MNIST and 96.62 with one epoch on Permuted-MNIST. These results demonstrate that memoryless approaches such as the one proposed here can achieve performances on par with the memory-replay-based approaches used as benchmarks, without the additional memory overhead. This suggests the need to identify more challenging continual learning assays.

Our approach allowed us to explore the transferability of optimal learning conditions across datasets and tasks, in order to understand the interplay between task-heterogeneity and continual learning across very different tasks. We demonstrated through systematic experiments that the accuracy of this transfer metalearning to datasets previously not seen can be largely explained through a transfer coefficient that can be based on metrics of dimensionality and distance between datasets.

Acknowledgements

This work was supported through the Lifelong Learning Machines (L2M) program from DARPA/MTO. The material is also based in part by work supported by the U.S. Department of Energy, Office of Science, under contract DE-AC02-06CH11357. We gratefully acknowledge the computing resources provided on Bebop (and/or Blues), a high-performance computing cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory.

References

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

Balaprakash, P., Salim, M., Uram, T., Vishwanath, V., and Wild, S. Deephyper: Asynchronous hyperparameter search for deep neural networks. In *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, pp. 42–51, Dec 2018. doi: 10.1109/HiPC.2018.00014.

Beaulieu, S., Frati, L., Miconi, T., Lehman, J., Stanley, K. O., Clune, J., and Cheney, N. Learning to continually learn. *arXiv preprint arXiv:2002.09571*, 2020.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Dayan, P. *Theoretical neuroscience : computational and mathematical modeling of neural systems*. Computational neuroscience. Massachusetts Institute of Technology Press, Cambridge, Mass., 2001. ISBN 0262041995.

Farquhar, S. and Gal, Y. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Hige, T., Aso, Y., Modi, M. N., Rubin, G. M., and Turner, G. C. Heterosynaptic plasticity underlies aversive olfactory learning in ¡em¿drosophila¡/em¿. *Neuron*, 88 (5):985–998, 2020/07/05 2015. doi: 10.1016/j.neuron.2015.11.003. URL https://doi.org/10.1016/j.neuron.2015.11.003.

Hsu, Y.-C., Liu, Y.-C., Ramasamy, A., and Kira, Z. Reevaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018.

Javed, K. and White, M. Meta-learning representations for continual learning. In *Advances in Neural Information Processing Systems*, pp. 1818–1828, 2019.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Litwin-Kumar, A., Harris, K. D., Axel, R., Sompolinsky, H., and Abbott, L. F. Optimal degrees of synaptic connectivity. *Neuron*, 93(5):1153–1164.e7, 2020/06/10 2017. doi: 10.1016/j.neuron.2017.01.030. URL https://doi.org/10.1016/j.neuron.2017.01.030.

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

Madireddy, S., Yanguas-Gil, A., and Balaprakash, P. Neuromorphic architecture optimization for task-specific dynamic learning. In *Proceedings of the International Conference on Neuromorphic Systems*, pp. 1–5, 2019.

Miconi, T., Clune, J., and Stanley, K. O. Differentiable plasticity: training plastic neural networks with backpropagation. *CoRR*, abs/1804.02464, 2018. URL http://arxiv.org/abs/1804.02464.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.

Oja, E. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982. doi: 10.1007/BF00275687. URL https://doi.org/10.1007/BF00275687.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

Rajasegaran, J., Khan, S., Hayat, M., Khan, F. S., and Shah, M. itaml: An incremental task-agnostic meta-learning approach. *arXiv preprint arXiv:2003.11652*, 2020.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

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 *International Conference on Machine Learning*, pp. 4528–4537, 2018.

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

Thrun, S. and Pratt, L. *Learning to learn*. Springer Science & Business Media, 2012.

van de Ven, G. M. and Tolias, A. S. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. https://github.com/zalandoresearch/fashion-mnist.git, 2017a.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017b.

Yanguas-Gil, A. Memristor design rules for dynamic learning and edge processing applications. *APL Materials*, 7(9):091102, 2019. doi: 10.1063/1.5109910. URL https://doi.org/10.1063/1.5109910.

Yanguas-Gil, A., Mane, A., Elam, J. W., Wang, F., Severa, W., Daram, A. R., and Kudithipudi, D. The insect brain as a model system for low power electronics and edge processing applications. In *2019 IEEE Space Computing Conference (SCC)*, pp. 60–66, 2019.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3987–3995. JMLR. org, 2017.