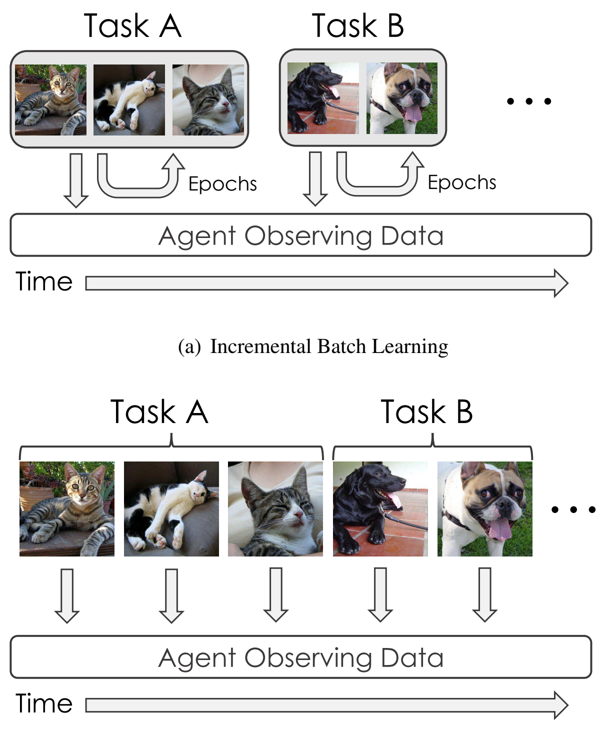
New Metrics and Experimental Paradigms for Continual Learning

Tyler L. Hayes Ronald Kemker Nathan D. Cahill Christopher Kanan
Rochester Institute of Technology
{tlh6792, rmk6217, ndcsma, kanan}@rit.edu

Abstract

In order for a robotic agent to learn successfully in an uncontrolled environment, it must be able to immediately alter its behavior. Deep neural networks are the dominant approach for classification tasks in computer vision, but typical algorithms and architectures are incapable of immediately learning new tasks without catastrophically forgetting previously acquired knowledge. There has been renewed interest in solving this problem, but there are limitations to existing solutions, including poor performance compared to offline models, large memory footprints, and learning slowly. In this abstract, we formalize the continual learning paradigm and propose new benchmarks for assessing continual learning agents.



(a) Incremental Batch Learning

(b) Continual Learning

Figure 1. **(a)** In incremental learning, an agent learns from non-iid batches of data containing a particular task, e.g., a single class in incremental class learning. The agent may observe the batch until it finishes learning it, but subsequently, it will never see that data again. In this example, each batch is denoted by a gray box containing a single class/task. **(b)** Conversely, in continual learning, as defined here, an agent is required to immediately learn non-iid data streams sample-by-sample, and the agent only has one look at each example. Continual learning more closely matches animal learning, and is required for deployed agents that must learn immediately.

1. Introduction

When working in uncontrolled environments, robots must quickly alter their behavior to learn and adapt in real-time. Deep neural networks (DNNs) are the current state-of-the-art method for machine perception, but they are incapable of learning new instances immediately. Learning requires looping over a dataset, which requires a considerable amount of time. Moreover, if streams of instances are not independent and identically distributed (iid), then a conventional DNN will suffer from catastrophic forgetting of previously learned information [4]. In contrast, a robot frequently may encounter non-iid streams of labeled data, e.g., when it is learning to recognize a particular object in its environment. In **continual learning**, sometimes known as *streaming learning*, an algorithm must be able to immediately make inferences from new examples, and must have the ability to learn from non-iid data streams. Here, we formalize the continual learning paradigm and describe appropriate performance metrics.

As shown in Fig. 1, we distinguish between incremental batch learning and continual learning. In incremental batch learning, a learner sequentially observes a labeled training dataset D broken up into T individual batches that cannot be assumed to be iid , i.e., $D = \bigcup_{t=1}^{T} B_t$. Each batch B_t consists of N_t labeled training data points, i.e., $B_t = \{(\mathbf{x}_i, k_i)\}_{i=1}^{N_t}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a training sample and $k_i \in C$ is a discrete label. The model is only permitted to learn from batches sequentially, in order, i.e., at time t it only has access to B_t. This paradigm is popular in the literature, and it was used to evaluate many recent algorithms, e.g., EWC [9], iCaRL [12], PathNet [3], FearNet [7], etc. Continual learning, as defined here, is incremental learning with an additional constraint that $N_t = 1$ (each batch contains a single example), each batch may only be observed once (a single epoch), and the model may be evaluated at any time. None of the aforementioned algorithms can operate in this continual learning paradigm.

In a real-world setting, robots must be capable of adapting to their environment quickly, efficiently, and reliably. It is unrealistic to provide an agent with every scenario it may encounter in an offline setting since this would require a significant amount of training time and data. Ideally, an agent would be capable of learning about, and adapting to, changes in its environment in real-time. These changes include appearance changes of objects or background scenery (e.g., seasonal changes) or examining new classes of objects that had not been previously observed. Continual learning addresses exactly these problems by forcing an agent to learn on a sample-by-sample basis in real-time, while also not catastrophically forgetting previously learned information. An agent is implicitly required to use its existing knowledge to make inferences about new situations and environments.

Continual learning is analogous to how animals learn and use knowledge, i.e., training examples must be learned sequentially (one-by-one), they are not assumed to be iid, there is no guarantee that an example can be observed more than once, the learner can be tested at any time, and memory resources must be independent of the size of the training dataset. Creating models capable of overcoming these constraints is necessary for developing advanced algorithms in embedded agents and robotics that must learn in real-time and are often resource constrained.

Continual learning more closely matches the requirements of a robotic learner. That is, the agent is only exposed to a single training example at any given time and only has one look at that example. Additionally, the agent must be capable of making inferences about its non-iid environment given existing knowledge, since it cannot simply store all previous training examples. For example, a robot may obtain multiple views (images) from a single instance of class A, then more views from another instance of class A, before finally learning instances from a class B, etc.

In this abstract, we describe experiments and metrics for testing a continual learner. These paradigms will enable new algorithms to be better compared and evaluated.

2. Evaluating Fast Continual Learning

2.1. Experimental Paradigms

We describe three paradigms for evaluating continual learning models: the data stream is completely unordered (iid), the data stream is ordered by class, and the data stream is temporally organized by instances. In all three paradigms, during training the model is required to learn on a sample-by-sample basis and is only allowed one epoch through the entire randomly sorted training set. It is evaluated every n samples and it does not know the value of n.

Learning iid Data Continually. The first continual learning experiment evaluates a continual learner's ability to learn quickly, without the need to compensate for non-iid data streams. While this is not a realistic scenario for a robot, this scenario should be the easiest for a continual learning model to rival an offline learner. It assumes that data is iid, with the data arriving in a randomly shuffled stream. In this scenario, it is typical for continual learning models to still perform worse than offline learners, making it a basic test for the model's abilities.

Learning Class Data Continually. The second paradigm tests the model's ability to learn new classes incrementally. This experiment will cause catastrophic forgetting in a conventional DNN, and even methods that are purported to be robust to catastrophic forgetting [8]. To assess model performance, test accuracy will be computed at regular intervals on data belonging to all previously observed classes. In this scenario classes are not revisited, which is the assumption that many other models make, e.g., iCaRL [12].

Learning Organized non-iid Data Continually. The final continual learning experiment measures each model's performance in the most realistic setting and differs significantly from previous training procedures. Data is ordered in batches from specific instances of particular categories and categories can be re-visited, e.g., 100 labeled images of dog #1, followed by 50 images of cat #2, followed by 200 images of cat #3, 83 labeled images of dog #4, etc. This is illustrated in Fig. 1(b). This scenario closely matches how a robot would experience stimuli, i.e., it would see multiple instances of a particular object and then it may not see that object class or instance again for a while. This scenario will cause catastrophic forgetting in conventional methods. To assess model performance, accuracy is computed at regular intervals on all test data.

2.2. Performance Metrics

Evaluating continual learning means evaluating the ability for a learner to learn quickly from non-iid data streams. It also means closely measuring the learner's memory usage, since one way to learn quickly is to simply store all training data as it is observed, which is impractical for a robot that is deployed for a long duration. In earlier work, Kemker et al. introduced new metrics for measuring memory stability, plasticity, and overall performance in incremental batch learning [8], and these metrics can be applied to continual learning. Overall performance of a continual learning method is given by:

$$\Omega_{\text{all}} = \frac{1}{T}\sum_{t=1}^{T}\frac{\alpha_{\text{all},t}}{\alpha_{\text{offline},t}} \ , \tag{1}$$

where $\alpha_{\text{all,t}}$ is the accuracy on all of the test data seen at test time t, $\alpha_{\text{offline,t}}$ is the accuracy of the optimized offline iid model on all of the training data until time t, and T denotes the total number of testing events. This metric enables an incrementally trained algorithm to be compared relative to an offline trained algorithm, with an $\Omega_{all} = 1$ indicating identical performance. Hypothetically, it is possible for $\Omega_{all} > 1$ if the offline model is worse than the one trained in the continual learning paradigm. For an offline learner, we recommend using heavily optimized and regularized state-of-the-art neural networks (e.g., ResNet [6]). In addition to the Ω_{all} metric, the amount of memory used by a model should also be reported as a function of time.

Since each of these experimental paradigms is formulated based on the organization of the data seen by the agent, we recommend running each experiment multiple times with different permutations of the dataset. The mean and standard deviation of the results over different permutations would then be reported to demonstrate a model's consistency and robustness to changes in ordering of the data.

2.3. Continual Learning Datasets

Two of the largest datasets for continual learning experiments are iCubWorld Transformations (iCub-T) [11] and CORe50 [10]. Both of these datasets are object recognition datasets designed specifically for continual learning with images generated from a sequence of frames of a person moving each object around. These datasets are ideal for evaluating continual learning because the data comes from particular instances in bursts while a robot is viewing that object. After seeing a burst of images from a particular instance, the agent then learns from another burst. It is naturally non-iid.

Both of these datasets only contain tens of classes of common household objects, which demonstrates a lack of size and diversity. To push the forefront of continual learning technology, we argue that improvements must be made to the existing continual learning datasets to make the training and testing of agents more robust and generalizable.

3. Baseline Experiments

We evaluated four continual learning algorithms: Incremental 1-Nearest Neighbor (1NN), biased ARTMAP (bARTMAP) [1], GeppNet [5], and an online multi-layer perceptron (MLP) by comparing them to a small offline MLP neural network. Results for the three paradigms on the small iCub World 1.0 dataset [2] using ResNet-50 [6] embeddings are shown in Table 1. No algorithm reaches the performance of the ideal learner, even on this easy dataset, demonstrating the difficulty the continual learning problem poses for existing models.

Method	**iid**	**class**	**organized non-iid**
MLP	**0.881**	0.308	0.255
1NN	0.836	0.894	**0.863**
bARTMAP [1]	0.787	**0.898**	0.800
GeppNet [5]	0.832	0.757	0.694
Offline (Ideal)	1.000	1.000	1.000

Table 1. Ω_{all} metrics computed for each of the streaming classification experiments on iCubWorld 1.0 [2]. Note that bARTMAP and GeppNet are not conventional DNNs.

4. Conclusion & Open Questions

While DNNs have become increasingly popular for solving a wide variety of problems, there are still many limitations to using these systems. Modern DNNs are known to suffer from catastrophic forgetting and cannot be trained in a continual learning framework for ease of use on embedded platforms. In this abstract, we provided explicit definitions for, and defined the difference between, incremental batch learning and continual learning. Additionally, we introduced three continual learning experimental paradigms and made recommendations for existing datasets and evaluation metrics that could be used to evaluate continual learning agents.

One major issue in continual learning is that datasets are too small. It is critical for future work to make datasets larger (hundreds to thousands of classes) and more diverse (different types of recognition, e.g., face, scene, activity). One interesting idea would be to create a dataset that contains classes/tasks in the test set that a model has never seen before. A model should then be able to indicate that it has not seen that class before.

While the iCub-T and CORe50 datasets are good for proof of concept models, results on these datasets are unlikely to indicate performance in a natural environment. For now, we recommend using iCub-T or CORe50, but ideally continual learning agents should be validated on much larger and diverse datasets.

Overcoming the constraints of continual learning would encourage the development of agents necessary for improving robotic vision. These capabilities would allow agents to learn from non-iid, temporally organized data streams, continuously learn and adapt to changes over time, and would have improved computational and memory efficiency.

References

[1] G. A. Carpenter and S. C. Gaddam. Biased art: a neural architecture that shifts attention toward previously disregarded features following an incorrect prediction. *Neural Networks*, 23(3):435–451, 2010.

[2] S. Fanello, C. Ciliberto, M. Santoro, L. Natale, G. Metta, L. Rosasco, and F. Odone. iCub World: Friendly robots help building good vision data-sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 700–705, 2013.

[3] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

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

[5] A. Gepperth and C. Karaoguz. A bio-inspired incremental learning architecture for applied perceptual problems. *Cognitive Computation*, 8(5):924–934, 2016.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] R. Kemker and C. Kanan. Fearnet: Brain-inspired model for incremental learning. In *International Conference on Learning Representations (ICLR)*, 2018.

[8] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan. Measuring catastrophic forgetting in neural networks. In *AAAI*, 2018.

[9] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, pages 3521–3526, 2017.

[10] V. Lomonaco and D. Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning (CoRL)*, 2017.

[11] G. Pasquale, C. Ciliberto, L. Rosasco, and L. Natale. Object identification from few examples by improving the invariance of a deep convolutional neural network. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4904–4911, 2016.

[12] S.-A. Rebuffi, A. Kolesnikov, and C. H. Lampert. icarl: Incremental classifier and representation learning. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2017.