The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)

Generative Continual Concept Learning

Mohammad Rostami
University of Pennsylvania
mrostami@seas.upenn.edu

Soheil Kolouri
HRL Laboratories, LLC
skolouri@hrl.com

James McClelland
Stanford University
jlmcc@stanford.edu

Praveen Pilly
HRL Laboratories, LLC
pkpilly@hrl.com

Abstract

After learning a concept, humans are also able to continually generalize their learned concepts to new domains by observing only a few labeled instances without any interference with the past learned knowledge. In contrast, learning concepts efficiently in a continual learning setting remains an open challenge for current Artificial Intelligence algorithms as persistent model retraining is necessary. Inspired by the Parallel Distributed Processing learning and the Complementary Learning Systems theories, we develop a computational model that is able to expand its previously learned concepts efficiently to new domains using a few labeled samples. We couple the new form of a concept to its past learned forms in an embedding space for effective continual learning. Doing so, a generative distribution is learned such that it is shared across the tasks in the embedding space and models the abstract concepts. This procedure enables the model to generate pseudo-data points to replay the past experience to tackle catastrophic forgetting.

Introduction

An important ability of humans is to continually build and update abstract concepts. Humans develop and learn abstract concepts to characterize and communicate their perception and ideas (Lake, Salakhutdinov, and Tenenbaum 2015). These concepts often are evolved and expanded efficiently as more experience about new domains is gained. Consider for example, the concept of the printed character "4". This concept is often taught to represent the "natural number four" in the mother tongue of elementary school students, e.g., English. Upon learning this concept, humans can efficiently expand it by observing only a few samples from other related domains, e.g., variety of hand written digits or printed digits in other secondary languages. Despite remarkable progress in Artificial intelligence (AI) over the past decade, learning concepts efficiently in a way similar to humans remains an unsolved challenge for AI. This is because the exceptional progress of AI is mostly driven by re-emergence of deep neural networks. Since deep networks are trained in an end-to-end supervised learning setting, access to labeled data is necessary for learning any new distribution. For this reason and despite emergence of behaviors similar to the nervous system in deep nets (Morgenstern, Rostami, and Purves 2014), adapting a deep neural network to learn a concept in a new domain usually requires model retraining from scratch which is conditioned on the availability of a large number of labeled samples in the new domain. Moreover, training deep networks in a continual learning setting is challenging due to the phenomenon of "catastrophic forgetting" (French 1999). When a network is trained on sequential tasks, the new learned knowledge usually interferes with past learned knowledge, causing forgetting what has been learned before.

In this paper, we develop a computational model that is able to expand and generalize learned concepts efficiently to new domains using a few labeled data from the new domains. We rely on Parallel Distributed Processing (PDP) paradigm (McClelland et al. 1986) for this purpose. Work on semantic cognition within the PDP framework hypothesizes that abstract semantic concepts are formed in higher level layers of the nervous system (McClelland and Rogers 2003; Saxe, McClelland, and Ganguli 2019). We model this hypothesis by assuming that the data points are mapped into an embedding space, which captures existing concepts. To prevent catastrophic forgetting, we rely on the Complementary Learning Systems (CLS) theory (McClelland, McNaughton, and O'Reilly 1995). CLS theory hypothesizes that continual lifelong learning ability of the nervous system is a result of a dual long- and short-term memory system. The hippocampus acts as short-term memory and encodes recent experiences that are used to consolidate the knowledge in the neocortex as long-term memory through offline experience replays during sleep (Diekelmann and Born 2010). This suggests that if we store suitable samples from past domains in a memory buffer, like in the neocortex, these samples can be replayed along with current task samples from recent-memory hippocampal storage to train the base model jointly on the past and the current experiences to tackle catastrophic forgetting.

More specifically, we model the latent embedding space via responses of a hidden layer in a deep neural network. Our idea is to stabilize and consolidate the data distribution in this space, where domain-independent abstract concepts are encoded. Doing so, new forms of concepts can be learned efficiently by coupling them to their past learned forms in

the embedding space. Data representations in this embedding space can be considered as neocortical representations in the brain, where the learned abstract concepts are captured. We model concept learning in a sequential task learning framework, where learning concepts in each new domain is considered to be a task. To generalize the learned concepts without forgetting, we use an autoencoder as the base network to benefit from efficient coding ability of deep autoencoders and model the embedding space as the middle layer of the autoencoder. This will also make our model generative, which can be used to implement the offline memory replay process in the sleeping brain (Rasch and Born 2013). To this end, we fit a parametric multi-modal distribution to the training data representations in the embedding space. The drawn points from this distribution can be used to generate pseudo-data points through the decoder network for experience replay to prevent catastrophic forgetting. We demonstrate that this learning procedure enables the base model to generalize its learned concepts to new domains using a few labeled samples.

Related Work

Lake et al. (Lake, Salakhutdinov, and Tenenbaum 2015) modeled human concept learning within a "Bayesian probabilistic learning" (BPL) paradigm. They present BPL as an alternative for deep learning to mimic the learning ability of humans as these models require considerably less amount of training data. The concepts are represented as probabilistic programs that can generate additional instances of a concept given a few samples of that concept. However, the proposed algorithm in Lake et al. (Lake, Salakhutdinov, and Tenenbaum 2015), requires human supervision and domain knowledge to tell the algorithm how the real-world concepts are generated. This approach seems feasible for the recognition task that they have designed to test their idea, but it does not scale to other more challenging concept learning problems. Our framework similarly relies on a generative model that can produce pseudo-samples of the learned concepts, but we follow an end-to-end deep learning scheme that automatically encodes concepts in the hidden layer of the network with minimal human supervision requirement. Our approach can be applied to a broader range of problems. The price is that we rely on data to train the model, but only a few data points are labeled. This is similar to humans with respect to how they too need practice to generate samples of a concept when they do not have domain knowledge (Longcamp, Zerbato-Poudou, and Velay 2005). This generative strategy has been used in the Machine Learning (ML) literature to address "few-shot learning" (FSL) (Snell, Swersky, and Zemel 2017; Motiian et al. 2017). The goal of FSL is to adapt a model that is trained on a source domain with sufficient labeled data to generalize well on a *related* target domain with a few target labeled data points. In our work, the domains are different but also are related in that they share similar concepts.

Most FSL algorithms consider only one source and one target domain, which are learned jointly. Moreover, the main goal is to learn the target task. In contrast, we consider a continual learning setting in which the domain-specific tasks arrive sequentially. Hence, catastrophic forgetting becomes a major challenge. An effective approach to tackle catastrophic forgetting is to use experience replay (McCloskey and Cohen 1989; Robins 1995). Experience replay addresses catastrophic forgetting via storing and replaying data points of past learned tasks continually. Consequently, the model retains the probability distributions of the past learned tasks. To avoid requiring a memory buffer to store past task samples, generative models have been used to produce pseudo-data points for past tasks. To this end, generative adversarial learning can be used to match the cumulative distribution of the past tasks with the current task distribution to allow for generating pseudo-data points for experience replay (Shin et al. 2017). Similarly, autoencoder structure can also be used to generate pseudo-data points (Parisi et al. 2019; Rostami, Kolouri, and Pilly 2019). Building upon our prior work (Rostami, Kolouri, and Pilly 2019), we develop a new method for generative experience replay to tackle catastrophic forgetting. Although prior works require access to labeled data for all the sequential tasks for experience replay, we demonstrate that experience replay is feasible even in the setting where only the initial task has labeled data. Our contribution is to combine ideas of few-shot learning with generative experience replay to develop a framework that can continually update and generalize learned concepts when new domains are encountered in a lifelong learning setting. We couple the distributions of the tasks in the middle layer of an autoencoder and use the shared distribution to expand concepts using a few labeled data points without forgetting.

Problem Statement and the Proposed Solution

In our framework, learning concepts in each domain is considered to be classes of an ML task, e.g., different types of digit characters. We consider a continual learning setting (Ruvolo and Eaton 2013), where an agent receives consecutive tasks $\{\mathcal{Z}^{(t)}\}_{t=1}^{T_{\text{Max}}}$ in a sequence $t = 1, \ldots, T_{\text{Max}}$ over its lifetime. The total number of tasks, distributions of the tasks, and the order of tasks is not known a priori. Since the agent is a lifelong learner, the current tasks is learned at each time step and the agent then proceeds to learn the next task. The knowledge that is gained from experiences is used to learn the current task efficiently, i.e., using minimal number of labeled data. The new learned knowledge from the current task also would be accumulated to the past experiences to potentially ease learning in future. Additionally, this accumulation must be done consistently to generalize the learned concepts as the agent must perform well on all learned task, i.e., not to forget. This is because the learned tasks may be encountered at any time in future. Figure 1 presents a high-level block-diagram visualization of this framework.

We model an abstract concept as a class within a domain-dependent classification task. Data points for each task t, are drawn i.i.d. from the joint probability distribution, i.e., $(\boldsymbol{x}_i^{(t)}, \boldsymbol{y}_i^{(t)}) \sim p^{(t)}(\boldsymbol{x}, \boldsymbol{y})$ which has the marginal distribution $q^{(t)}(\boldsymbol{x})$ over $\boldsymbol{x}$. We consider a deep neural network $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^k$ as the base learning model, where θ denote the learnable weight parameters. A deep network is able to solve classification tasks through extracting task-dependent high quality features in a data-driven end-to-end learning (Krizhevsky, Sutskever, and Hinton 2012). Within

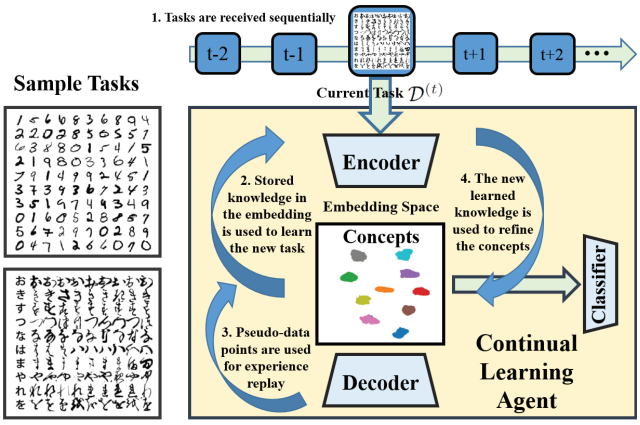


Figure 1: Architecture of the proposed framework.

PDP paradigm (McClelland et al. 1986; McClelland and Rogers 2003; Saxe, McClelland, and Ganguli 2019), this means that the data points are mapped into a discriminative embedding space, modeled by the network hidden layers, where the classes become separable, i.e., data points belonging to a class are grouped as an abstract concept. On this basis, the deep network f_θ is a functional composition of an encoder $\phi_{\boldsymbol{v}}(\cdot) : \mathbb{R}^d \rightarrow \mathcal{Z} \subset \mathbb{R}^f$ with learnable parameter $\boldsymbol{v}$, that encode the input data into the embedding space $\mathcal{Z}$ and a classifier sub-network $h_{\boldsymbol{w}}(\cdot) : \mathbb{R}^f \rightarrow \mathbb{R}^k$ with learnable parameters $\boldsymbol{w}$, that maps encoded information into the label space. In other words, the encoder network changes the input data distribution as a deterministic function. Because the embedding space is discriminative, data distribution in the embedding space would be a multi-modal distribution that can be modeled as Gaussian mixture model (GMM). Figure 1 visualizes this intuition based on experimental data, used in the experimental validation section.

Within ML formalism, the agent can solve the task $\mathcal{Z}^{(1)}$ using standard empirical risk minimization (ERM). Given the labeled training dataset $\mathcal{D}^{(1)} = \langle \boldsymbol{X}^{(1)}, \boldsymbol{Y}^{(1)} \rangle$, where $\boldsymbol{X}^{(1)} = [\boldsymbol{x}_1^{(1)}, \ldots, \boldsymbol{x}_{n_t}^{(1)}] \in \mathbb{R}^{d \times n_1}$ and the labeles $\boldsymbol{Y}^{(1)} = [\boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_n^{(1)}] \in \mathbb{R}^{k \times n_t}$, we can solve for the network optimal weight parameters: $\hat{\theta}^{(t)} = \arg\min_\theta \hat{e}_\theta = \arg\min_\theta 1/n_t \sum_i \mathcal{L}_d(f_\theta(\boldsymbol{x}_i^{(t)}), \boldsymbol{y}_i^{(t)})$. Here, $\mathcal{L}_d(\cdot)$ is a suitable loss function, e.g., cross entropy. Conditioned on having large enough number of labeled data points n_1, the empirical risk would be a suitable function to estimate the real risk function, $e = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p^{(t)}(\boldsymbol{x},\boldsymbol{y})}(\mathcal{L}_d(f_{\theta^{(t)}}(\boldsymbol{x}), \boldsymbol{y}))$ (Shalev-Shwartz and Ben-David 2014) as the Bayes optimal objective. Hence, the trained model will generalize well on test data points for the task $\mathcal{Z}^{(1)}$. Good generalization performance means that each class would be learned as a concept which is encoded in the hidden layers. Our goal is to consolidate these learned concepts and generalize them when the next tasks with minimal labeled data arrive. That is, for tasks $\mathcal{Z}^{(t)}, t > 1$, we have access to the dataset $\mathcal{D}^{(t)} = \langle \{\boldsymbol{X}'^{(t)}, \boldsymbol{Y}^{(t)}\}, \boldsymbol{X}^{(t)} \rangle$, where $\boldsymbol{X}'^{(t)} \in \mathbb{R}^{d \times n_t}$ denotes the labeled data points and $\boldsymbol{X}^{(t)} \in \mathbb{R}^{d \times n_t}$ denotes unlabeled data points. This learning setting means that the learned concepts must be generalized in the subsequent domains with minimal supervision. Standard ERM can not be used to learn the subsequent tasks because the number of labeled data points is not sufficient, i.e., overfitting would occur. Additionally, even in the presence of enough labeled data, catastrophic forgetting would be consequence of using ERM. This is because the model parameters will be updated using solely the current task data which can potentially deviate the values of $\theta^{(T)}$ from the previous learned values in the past time step. Hence, the agent would not retain its learned knowledge.

Following PDP hypothesis, our goal is to use the encoded distribution in the embedding space to expand the concepts that are captured the embedding space such that catastrophic forgetting does not occur. The gist of our idea is to update the encoder sub-network such that each subsequent task is learned such that its distribution in the embedding space matches the distribution that is shared by $\{\mathcal{Z}^{(t)}\}_{t=1}^{T-1}$ at $t = T$. Since this distribution is initially learned via $\mathcal{Z}^{(1)}$ and subsequent tasks are enforced to share this distribution in the embedding space with $\mathcal{Z}^{(1)}$, we do not need to learn it from scratch as the concepts are shared across the tasks. Hence, since the embedding becomes invariant with respect to any learned input task, catastrophic forgetting would not occur.

The key challenge is to adapt the standard ERM such that the tasks share the same distribution in the embedding space becomes. To this end, we modify the base network $f_\theta(\cdot)$ to form a generative autoencoder by amending the model with a decoder $\psi_{\boldsymbol{u}} : \mathcal{Z} \rightarrow \mathcal{X}$.We train the model such the pair $(\phi_{\boldsymbol{u}}, \psi_{\boldsymbol{u}})$ form an autoencoder. Doing so, we enhance the ability of the model to encode the concepts as separable clusters in the embedding. We use the knowledge about data distribution form in the embedding to match the distributions of all tasks in the embedding. This leads to consistent generalization of the learned concepts. Additionally, since the model is generative and knowledge about past experiences is encoded in the network, we can use CLS process (McClelland, McNaughton, and O'Reilly 1995) to prevent catastrophic forgetting. When learning a new task, pseudo-data points for the past learned tasks can be generated by sampling from the shared distribution in the embedding and feeding the samples to the decoder sub-network. These pseudo-data points are used along with new task data to learn each task. Since the new task is learned such that its distribution matches the past shared distribution, pseudo-data points generated for learning future tasks would also represent the current task as well.

Proposed Algorithm

Following the above framework, learning the first task ($t = 1$) reduces to minimizing the discrimination loss for classification and the autoencoder reconstruction loss to solve for optimal parameters:

$$\min_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{u}} \mathcal{L}_c(\boldsymbol{X}^{(1)}, \boldsymbol{Y}^{(1)}) = \min_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{u}} \frac{1}{n_1} \sum_{i=1}^{n_1} \Bigg(\mathcal{L}_d\Big(h_{\boldsymbol{w}}(\phi_{\boldsymbol{v}}(\boldsymbol{x}_i^{(1)})), \boldsymbol{y}_i^{(1)}\Big) + \gamma \mathcal{L}_r\Big(\psi_{\boldsymbol{u}}(\phi_{\boldsymbol{v}}(\boldsymbol{x}_i^{(1)})), \boldsymbol{x}_i^{(1)}\Big) \Bigg), \tag{1}$$

where $\mathcal{L}_r$ is the reconstruction loss, $\mathcal{L}_c$ is the combined loss, and γ is a trade-off parameter.

If the base learning model is complex enough, the concepts would be formed in the embedding space as separable clusters upon learning the first task. This means that the data distribution can be modeled as a GMM distribution in the embedding. We can use standard methods such as expectation maximization to fit a GMM distribution with k components to the multimodal empirical distribution formed by the drawn samples $\{(\phi_{\boldsymbol{v}}(\boldsymbol{x}_i^{(1)}), \boldsymbol{y}_i^{(1)})_{i=1}^{n_1}\}_{i=1}^{n_1} \sim p^{(0)}$ in the embedding space. Let $\hat{p}_k^{(0)}(\boldsymbol{z})$ denote the estimated parametric GMM distribution with k components. The goal is to retain this initial estimation that captures concepts when future domains are encountered. Following PDP framework, we learn the subsequent tasks such that the current task shares the same GMM distribution with the previous learned tasks in the embedding space. We also update the estimate of the shared distribution after learning each subsequent task. Updating this distribution means generalizing the concepts to the new domains without forgetting the past domains. As a result, the distribution $\hat{p}_{J,k}^{(t-1)}(\boldsymbol{z})$ captures knowledge about past domains when $\mathcal{Z}^{(t)}$ is being learned. Moreover, we can perform experience replay by generating pseudo-data points by first drawing samples from $\hat{p}_{J,k}^{(t-1)}(\boldsymbol{z})$ and then passing the samples through the decoder sub-network. The remaining challenge is to update the model such that each subsequent task is learned such that its corresponding empirical distribution matches $\hat{p}_{J,k}^{(t-1)}(\boldsymbol{z})$ in the embedding space. Doing so, ensures suitability of GMM to model the empirical distribution.

To match the distributions, let $\mathcal{D}_{ER}^{(t)} = \langle \psi(\boldsymbol{Z}_{ER}^{(t)}), \boldsymbol{Y}_{ER}^{(t)} \rangle$ denote the pseudo-dataset for tasks $\{\mathcal{Z}^{(s)}\}_{s=1}^{t-1}$, generated for experience replay when $\mathcal{Z}^{(t)}$ is being learned. Following the described framework, we form the following optimization problem to learn $\mathcal{Z}^{(t)}$ and generalized concepts:

$$\begin{aligned}
&\min_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{u}} \mathcal{L}_c(\boldsymbol{X}'^{(t)}, \boldsymbol{Y}^{(t)}) + \mathcal{L}_c(\boldsymbol{X}_{(ER)}^T, \boldsymbol{Y}_{(ER)}^T) + \\
&\eta D\Big(\phi_{\boldsymbol{v}}(q^{(t)}(\boldsymbol{X}^{(t)})), \hat{p}_{J,k}^{(t)}(\boldsymbol{Z}_{ER}^{(T)})\Big) + \\
&\lambda \sum_{j=1}^{k} D\Big(\phi_{\boldsymbol{v}}(q^{(t)}(\boldsymbol{X}'^{(t)})|C_j), \hat{p}_{J,k}^{(t)}(\boldsymbol{Z}_{ER}^{(T)}|C_j)\Big), \quad \forall t \geq 2,
\end{aligned} \tag{2}$$

where $D(\cdot,\cdot)$ is a suitable metric function to measure the discrepancy between two probability distributions. λ and η are a trade-off parameters. The first two terms in Eq. (2) denote the combined loss terms for each of the current task few labeled data points and the generated pseudo-dataset, defined similar to Eq. (1). The third and the fourth terms implement our idea and enforce the distribution for the current task to be close to the distribution shared by the past learned task. The third term is added to minimize the distance between the distribution of the current tasks and $\hat{p}_{J,k}^{(t-1)}(\boldsymbol{z})$ in the embedding space. Data labels is not needed to compute this term. The fourth term may look similar but note that we have conditioned the distance between the two distribution on the concepts to avoid the matching challenge, i.e., when wrong concepts (or classes) across two tasks are matched in the embedding space (Globerson and Roweis 2006). We use the few labeled data that are accessible for the current task to compute this term. Adding these terms guarantees that we can continually use GMM to model the shared distribution in the embedding.

Algorithm 1 ECLA (L, λ, η)

Input: data $\mathcal{D}^{(1)} = (\boldsymbol{X}^{(1)}, \boldsymbol{Y}^{(t)})$.
  $\mathcal{D}^{(t)} = (\{\boldsymbol{X}'^{(t)}, \boldsymbol{Y}^{(t)}\}, \boldsymbol{X}^{(t)})$ for $t = 2, \ldots, T_{\text{Max}}$
Concept Learning: learning the first task ($t = 1$) by solving (1)
Fitting GMM:
estimate $\hat{p}_k^{(0)}(\cdot)$ using $\{\phi_{\boldsymbol{v}}(\boldsymbol{x}_i^{(1)}))\}_{i=1}^{n_t}$
for $t \geq 2$ **do**
  Generate the pseudo dataset:
  $\mathcal{D}_{\text{ER}} = \{(\boldsymbol{x}_{ER,i}^{(t)} = \psi(\boldsymbol{z}_{ER,i}^{(t)}), \boldsymbol{y}_{ER,i}^{(t)})\}$
  $(\boldsymbol{z}_{ER,i}^{(t)}, \boldsymbol{y}_{ER,i}^{(t)}) \sim \hat{p}_k^{(t-1)}(\cdot)$
  Update:
  learnable parameters are updated by
  solving Eq. (2)
  Concept Generalization:
  update $\hat{p}_k^{(t)}(\cdot)$ using the combined samples
  $\{\phi_{\boldsymbol{v}}(\boldsymbol{x}_i^{(t)}), \phi_{\boldsymbol{v}}(\boldsymbol{x}_{ER,i}^{(t)})\}_{i=1}^{n_t}$
end for

The main remaining question is selection of a suitable probability distance metric $D(\cdot,\cdot)$. Common probability distance measures such as Jensen–Shannon divergence KL divergence are not applicable for our problem as the gradient for these measures is zero when the corresponding distributions have non-overlapping supports (Rabin and Peyré 2011). Since deep learning optimization problems are solved using first-order gradient-based optimization methods, we must select a distribution metric which has non-vanishing gradients. For this reason, we select the Wasserstein Distance (WD) metric (Bonnotte 2013) which satisfies this requirement and has recently been used extensively in deep learning applications to measure the distance between two probability distributions (Courty et al. 2017). In particular, we use Sliced Wasserstein Distance (SWD) (Bonneel et al. 2015) which is a suitable approximation for WD, while it can be computed efficiently using empirical samples, drawn from two distributions. Our concept learning algorithm, Efficient Concept Learning Algorithm (ECLA), is summarized in Algorithm 1.

Theoretical Analysis

We follow a standard PAC-learning style framework to analyze our algorithm (Shalev-Shwartz and Ben-David 2014) and using result from domain adaptation (Redko, Habrard, and Sebban 2017) to demonstrate the effectiveness of our algorithm. We perform the analysis in the embedding space $\mathcal{Z}$, where the hypothesis class is the set of all the classifiers $h_{\boldsymbol{w}}(\cdot)$ parameterized by $\boldsymbol{w}$. For any given model h in this class, let $e_t(h)$ denotes the observed risk for the domain that contains the task $\mathcal{Z}^{(t)}$, $e_{t'}(h)$ denotes the observed

risk for the same model on another secondary domain, and $\boldsymbol{w}^*$ denotes the optimal parameter for training the model on these two tasks jointly, i.e., $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} e_{\mathcal{C}}(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}}\{e_t(h) + e_{t'}(h)\}$. We also denote the Wasserstein distance between two given distributions as $W(\cdot,\cdot)$. We rely on the following theorem (Redko, Habrard, and Sebban 2017) which relates performance of a model trained on a particular domain to another secondary domain.

Theorem 0.1. *Consider two tasks $\mathcal{Z}^{(t)}$ and $\mathcal{Z}^{(t')}$ with n_t and $n_{t'}$ training data points, respectively. Let $h_{\mathbf{w}^{(t')}}$ be a model trained for $\mathcal{Z}^{(t')}$, then for any $d' > d$ and $\zeta < \sqrt{2}$, there exists a constant number N_0 depending on d' such that for any $\xi > 0$ and $\min(n_t, n_{t'}) \geq N_0 \max(\xi^{-(d'+2)}, 1)$ with probability at least $1-\xi$ for all $f_{\theta^{(t')}}$, the following holds:*

$$e_t(h) - e_{t'}(h) \leq W(\hat{p}^{(t)}, \hat{p}^{(t')}) + e_{\mathcal{C}}(\boldsymbol{w}^*) + \sqrt{\big(2\log(\frac{1}{\xi})/\zeta\big)}\big(\sqrt{\frac{1}{n_t}} + \sqrt{\frac{1}{n_{t'}}}\big), \tag{3}$$

where $\hat{p}^{(t)}$ and $\hat{p}^{(t')}$ are empirical distributions formed by the drawn samples from $p^{(t)}$ and $p^{(t')}$.

Theorem 0.1 is a broad result that provides an upper-bound on performance degradation of a trained model, when used in another domain. It suggests that if the model performs well on $\mathcal{Z}^{(t')}$ and if the upper-bound is small, then the model performs well on $\mathcal{Z}^{(t')}$. The last term is a constant term which depends on the number of available samples. This term is negligible when $n_t, n_{t'} \gg 1$. The two important terms are the first and the second terms. The first term is the Wasserstein distance between the two distributions. It may seem that according to this term, if we minimize the WD between two distributions, then the model should perform well on $\mathcal{Z}^{(t)}$. But it is crucial to note that the upper-bound depends on the second term as well. The second term suggests that the base model should be able to learn both tasks jointly. However, in the presence of "XOR classification problem", the tasks cannot be learned by a single model (Mangal and Singh 2007). This means that not only the WD between two distributions should be small, but the distributions should be aligned class-conditionally. Building upon Theorem 0.1, we provide the following theorem for our framework.

Theorem 0.2. *Consider ECLA algorithm at learning time step $t = T$. Then all tasks $t < T$ and under the conditions of Theorem 0.1, we can conclude:*

$$e_t \leq e^J_{T-1} + W(\phi(\hat{q}^{(t)}), \hat{p}^{(t)}_{J,k}) + \sum_{s=t}^{T-2} W(\hat{p}^{(s)}_{J,k}, \hat{p}^{(s+1)}_{J,k}) + e_{\mathcal{C}}(\boldsymbol{w}^*) + \sqrt{\big(2\log(\frac{1}{\xi})/\zeta\big)}\big(\sqrt{\frac{1}{n_t}} + \sqrt{\frac{1}{n_{er,t-1}}}\big), \tag{4}$$

where e^J_{T-1} denotes the risk for the pseudo-task with the distribution $\psi(\hat{p}^{(T-1)}_{J,k})$.

Proof: In Theorem 0.1, consider the task $\mathcal{Z}^{(t)}$ with the distribution $\phi(q^{(t)})$ and the pseudo-task with the distribution $p^{(T-1)}_k$ in the embedding space. We can use the triangular inequality recursively on the term $W(\phi(\hat{q}^{(t)}), \hat{p}^{(T-1)}_{J,k})$ in Eq. (3), i.e., $W(\phi(\hat{q}^{(t)}), \hat{p}^{(s)}_{J,k}) \leq W(\phi(\hat{q}^{(t)}), \hat{p}^{(s-1)}_{J,k}) + W(\hat{p}^{(s)}_{J,k}, \hat{p}^{(s-1)}_{J,k})$ for all time steps $t \leq s < T$. Adding up all the terms, concludes Eq. (4).

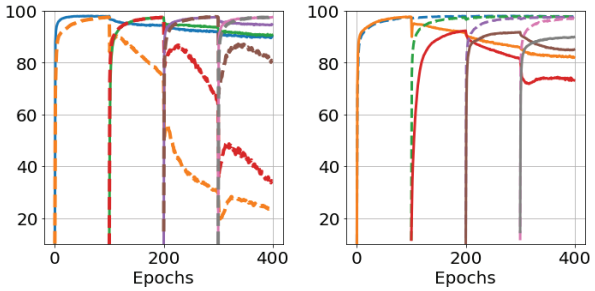
We can rely on Theorem 0.2 to demonstrate that why our algorithm can generalize concepts without forgetting the past learned knowledge. The first term in Eq. (4) is small because, experience replay minimizes this term using the labeled pseudo-data set via ERM. The fourth term is small since we use the few labeled data points to align the distributions class conditionally in Eq. (2). The last term is a negligible constant for $n_t, n_{er,t-1} \gg 1$. The second term denotes the distance between the task distribution and the fitted GMM. When the PDP hypothesis holds and the model learns a task well, this term is small as we can approximate $\phi(\hat{q}^{(t)})$ with $\hat{p}^{(s-1)}_{J,k}$) (see Ashtiani et al. (Ashtiani et al. 2018) for a rigorous analysis of estimating a distribution with GMM). In other words, this term is small if the classes are learned as concepts. Finally, the terms in the sum term in Eq 4 are minimized because at $t = s$ we draw samples from $p^{(s-1)}_k$ and by learning $\psi^{-1} = \psi$ enforce that $\hat{p}^{(s-1)}_{J,k} \approx \phi(\psi(\hat{p}^{(s-1)}_{J,k}))$. The sum term in Eq 4 models the effect of history. After learning a task and moving forward, this term potentially grows as more tasks are learned. This means that forgetting effects would increase as more subsequent tasks are learned which is intuitive. To sum up, ECLA minimizes the upper bound of e_t in Eq 4. This means that the model can learn and remember $\mathcal{Z}^{(t)}$ which in turn means that the concepts have been generalized without being forgotten on the old domains.

Experimental Validation

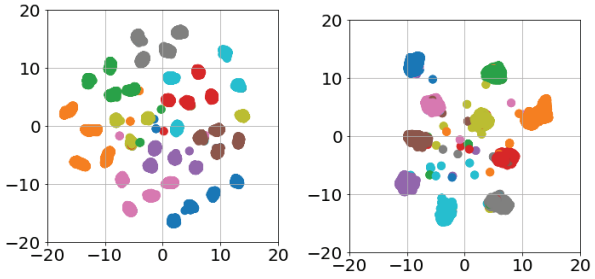
We validate our method on learning two sets of sequential learning tasks: permuted MNIST tasks and digit recognition tasks. These are standard benchmark classification tasks for sequential task learning. We adjust them for our learning setting. Each class in these tasks is considered to be a concept, and each task of the sequence is considered to be learning the concepts in a new domain.

Learning permuted MNIST tasks

Permuted MNIST tasks is standard benchmark that is designed for testing abilities of AI algorithms to overcome catastrophic forgetting (Shin et al. 2017; Kirkpatrick et al. 2017). The sequential tasks are generated using the MNIST ($\mathcal{M}$) digit recognition dataset (LeCun et al. 1990). Each task in the sequence is generated by applying a fixed random shuffling to the pixel values of digit images across the MNIST dataset (Kirkpatrick et al. 2017). As a result, generated tasks are homogeneous in terms of difficulty and are suitable to perform controlled experiments. Our learning setting is different compared to prior works as we considered the case where only the data for the initial MNIST task is fully labeled. In the subsequent tasks, only few data points are labeled. To the best of our knowledge, no precedent method addresses

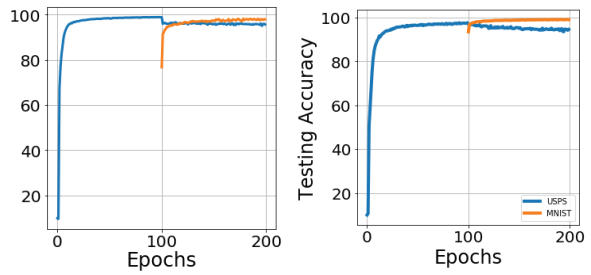


(a) BP vs. CLEER (b) ECLA vs. FR

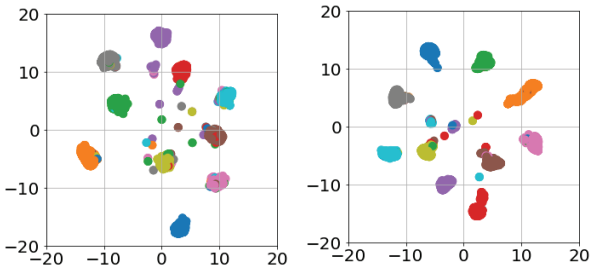


(c) FR (d) ECLA

Figure 2: Learning curves for four permuted MNIST tasks((a) and (b)) and UMAP visualization of ECLA vand FR in the embedding ((c) and (d)). (Best viewed in color.)



(a) $\mathcal{M} \rightarrow \mathcal{U}$ (b) $\mathcal{U} \rightarrow \mathcal{M}$



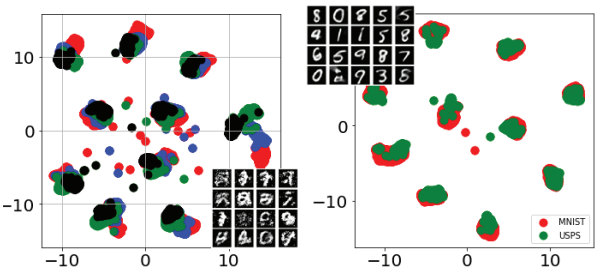
(c) $\mathcal{M} \rightarrow \mathcal{U}$ (d) $\mathcal{U} \rightarrow \mathcal{M}$

Figure 3: Performance results on MNIST and USPS digit recognition tasks ((a) and (b)). UMAP visualization for $\mathcal{M} \rightarrow \mathcal{U}$ and $\mathcal{U} \rightarrow \mathcal{M}$ tasks ((c) and (d)). (Best viewed in color.)

this learning scenario for direct comparison, so we only compared against: a) classic back propagation (BP) single task learning, (b) full experience replay (FR) using full stored data for all the previous tasks, and (c) learning using fully labeled data (CLEER) (Rostami, Kolouri, and Pilly 2019). We use the same base network structure for all the methods for fair comparison. BP is used to demonstrate that our method can address catastrophic forgetting as a lower-bound. FR is used as absolute an upper-bound to demonstrate that our method is able to learn cross-task concepts without using fully labeled data. CLEER is an instance of ECLA where fully labeled data is used to learn the subsequent tasks. We used CLEER to compare our method against an upper-bound.

We used standard stochastic gradient descent to learn the tasks and created learning curves by computing the performance of the model on the standard testing split of the current and the past learned tasks at each learning iteration. Figure 2 presents learning curves for four permuted MNIST tasks. Figure 2a presents learning curves for BP (dashed curves) and CLEER (solid curves). As can be seen, CLEER (i.e., ECLA with fully labeled data) is able to address catastrophic forgetting. This figure demonstrates that our method can be used as a new algorithm on its own to address catastrophic forgetting using experience replay (Shin et al. 2017). Figure 2b presents learning curves for FR (dashed curves) and ECLA (solid curve) when 5 labeled data points per class are used respectively. We observe that FR can tackle catastrophic forgetting perfectly but the challenge is the memory buffer requirement, which grows linearly with the number of learned tasks, making this method only suitable for comparison as an upper-bound. FR result also demonstrates that if we can generate high-quality pseudo-data points, catastrophic forgetting can be prevented completely. Deviation of the pseudo-data from the real data is the major reason for the initial performance degradation of ECLA on all the past learned tasks, when a new task arrives and its learning starts. This degradation can be ascribed to the existing distance between $\hat{p}_{J,k}^{(T-1)}$ and $\phi(q^{(s)})$ at $t = T$ for $s < T$. Note also as our theoretical analysis predicts, the performance on a past learned task degrades more as more tasks are learned subsequently. This is compatible with the nervous system as memories fade out as time passes unless enhanced by continually experiencing a task or a concept.

In addition to requiring fully labeled data, we demonstrate that FR does not identify concepts across the tasks. To this end, we have visualized the testing data for all the tasks in the embedding space $\mathcal{Z}$ in Figures 2 for FR and ECLA after learning the fourth task. For visualization purpose, we have used UMAP (McInnes, Healy, and Melville 2018), which reduces the dimensionality of the embedding space to two. In Figure 2c and Figure 2d, each color denotes the data points of one of the digits $\{0, 1, \ldots, 9\}$ (each circular shape indeed is a cluster of data points). We can see that the digits form separable clusters for both methods. This result is consistent with the PDP hypothesis and is the reason behind good performance of both methods. It also demonstrates why GMM is a suitable selection to model the data distribution in the embedding space. However, we can see that when FR is used, four distinct clusters for each digit are formed (i.e., one clus-



(a) Permuted MNIST tasks (b) $\mathcal{U} \rightarrow \mathcal{M}$

Figure 4: UMAP visualizations of data representations for (a) $\mathcal{U} \rightarrow \mathcal{M}$ and (b) permuted MNIST tasks in the embedding along with a few generated pseudo-data points after learning the final task. (Best viewed in color.)

ter per domain for each digit class). In other words, FR is unable to identify and generalize abstract concepts across the domains. In contrast, we have exactly ten clusters for the ten digits when ECLA is used, and hence the concepts are identified across the domains. This is the reason that we can generalize the learned concepts to new domains, despite using few labeled data.

Learning sequential digit recognition tasks

We performed a second set of experiments on a more realistic scenario. We consider two handwritten digit recognition datasets for this purpose: MNIST ($\mathcal{M}$) and USPS ($\mathcal{U}$) datasets. USPS dataset is a more challenging classification task as the size of the training set is smaller (20,000 compared to 60,000 images). We performed experiments on the two possible sequential learning scenarios $\mathcal{M} \rightarrow \mathcal{U}$ and $\mathcal{U} \rightarrow \mathcal{M}$. The experiments can be considered as concept learning for numeral digits as both tasks are digit recognition tasks but in different domains, i.e. written by different people.

Figure 3a and Figure 3b present learning curves for these two tasks when 10 labeled data points per class are used for the training of the second task. Note that the network retains the knowledge about the first task quite well following the learning of the second task. As expected from the theoretical justification, this empirical result suggests the performance of our algorithm depends on closeness of the distribution $\psi(\hat{p}^{(t)}_{J,k})$ to the distributions of previous tasks, and improving probability estimation will boost the performance of our approach. Since the two tasks are much more related compared to the permuted MNIST task, forming abstract concepts is more feasible. Additionally, we see a jumstart in the performance for the second task in both Figure 3a and Figure 3b which demonstrates knowledge transfer from the previous task due to the inherent similarity between the concepts. We have also presented UMAP visualization of the data points for the tasks in the embedding space in Figures 3c and Figures 3d. We observe that the distributions are matched in the embedding space and cross-domain concepts are learned by the network. These results demonstrate that our algorithm inspired by PDP and CLS theories can generalize concepts to new domains using few labeled data points.

Finally, to clarify how our approach is able to learn task-agnostic concepts, Figure 4a and Figure 4b present UMAP visualizations of data representations for $\mathcal{U} \rightarrow \mathcal{M}$ (related and similar tasks) and permuted MNIST tasks in the embedding, respectively. In these figures, data clusters for each task are shown with the same color. Interestingly, we observe two distinct phenomena in these two figures. When the tasks are visually similar (i.e., digit recognition tasks), data points for each class across the tasks are mixed in the embedding (in Figure 4b the green and red clusters almost completely overlap). This means that the corresponding cluster for each class in the embedding is task-agnostic. We have visualized a few random pseudo-data points that the model generated. As can be seen, with the exception of one data point, the generated data points are similar to real digits that can represent both MNIST and USPS datasets. We conclude that when the tasks are similar, our algorithm builds clusters that represent concepts (e.g., digit "4") that transcend the tasks and allow the model to generate pseudo-data points that can represent all tasks. In contrast, when the tasks are not very similar (i.e., permuted MNIST tasks), and we synthetically consider and enforce that they share the same classes, the formed clusters are structured but exhibit a different profile. In Figure 4a, it can be seen that the data points for the four tasks in each cluster are not completely mixed; i.e., the clusters are divided among the tasks. Our algorithm works because pseudo-data points for different tasks are generated from different task-specific regions of the clusters. Of course, the model will also generate pseudo-data points that are similar to combinations of data points of two or more tasks (i.e., when we sample from a region in the cluster that is shared among several tasks), but these data points do not harm learning or cause forgetting effect as they are consistent with the clusters. This can be seen by observing the generated pseudo-data points in Figure 4a carefully. We make an important conclusion that task-agnostic concepts can be abstracted for "continual concept learning" if there is an inherent similarity of the concepts across the tasks in the input space. Our algorithm also works for dissimilar tasks because we are synthetically enforcing the tasks to share the same classes, similar to the permuted MNIST tasks, but the resulting clusters would then not be completely task-agnostic. Interestingly, however, the clusters for each class across the dissimilar tasks are themselves clustered with respect to those for the other classes.

Conclusions

Inspired by the CLS theory and the PDP paradigm, we developed an algorithm that enables a deep network to update and generalize its learned concepts in a continual learning setting. The proposed algorithm is able to address the learning challenges by accumulating the new learned knowledge consistently to the past learned knowledge. For this purpose, our generative framework encodes abstract concepts in a hidden layer of the deep network in the form of a parametric GMM distribution. This distribution can be used to generalize concepts to new domains, where only a few labeled samples are accessible. Additionally, the model is able to generate pseudo-data points for past tasks, which can be used for experience replay to tackle catastrophic forgetting. Future work

will extend our model to detect new concepts automatically and actively ask for few labeled data points as unseen concept samples are encountered.

Acknowledgments

This material is based upon work supported by the United States Air Force and DARPA under Contract No. FA8750-18-C-0103. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force and DARPA.

References

Ashtiani, H.; Ben-David, S.; Harvey, N.; Liaw, C.; Mehrabian, A.; and Plan, Y. 2018. Nearly tight sample complexity bounds for learning mixtures of gaussians via sample compression schemes. In *Advances in Neural Information Processing Systems*, 3412–3421.

Bonneel, N.; Rabin, J.; Peyré, G.; and Pfister, H. 2015. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision* 51(1):22–45.

Bonnotte, N. 2013. *Unidimensional and evolution methods for optimal transportation*. Ph.D. Dissertation, Paris 11.

Courty, N.; Flamary, R.; Tuia, D.; and Rakotomamonjy, A. 2017. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence* 39(9):1853–1865.

Diekelmann, S., and Born, J. 2010. The memory function of sleep. *Nat Rev Neurosci* 11(114).

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences* 3(4):128–135.

Globerson, A., and Roweis, S. T. 2006. Metric learning by collapsing classes. In *Advances in neural information processing systems*, 451–458.

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

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

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science* 350(6266):1332–1338.

LeCun, Y.; Boser, B. E.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W. E.; and Jackel, L. D. 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, 396–404.

Longcamp, M.; Zerbato-Poudou, M.-T.; and Velay, J.-L. 2005. The influence of writing practice on letter recognition in preschool children: A comparison between handwriting and typing. *Acta psychologica* 119(1):67–79.

Mangal, M., and Singh, M. P. 2007. Analysis of multidimensional xor classification problem with evolutionary feedforward neural networks. *International Journal on Artificial Intelligence Tools* 16(01):111–120.

McClelland, J. L., and Rogers, T. T. 2003. The parallel distributed processing approach to semantic cognition. *Nature reviews neuroscience* 4(4):310.

McClelland, J. L.; Rumelhart, D. E.; Group, P. R.; et al. 1986. Parallel distributed processing. *Explorations in the Microstructure of Cognition* 2:216–271.

McClelland, J. L.; McNaughton, B. L.; and O'Reilly, R. C. 1995. 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.

McCloskey, M., and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24. Elsevier. 109–165.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Morgenstern, Y.; Rostami, M.; and Purves, D. 2014. Properties of artificial networks evolved to contend with natural spectra. *Proceedings of the National Academy of Sciences* 111(Supplement 3):10868–10872.

Motiian, S.; Jones, Q.; Iranmanesh, S.; and Doretto, G. 2017. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 6670–6680.

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

Rabin, J., and Peyré, G. 2011. Wasserstein regularization of imaging problem. In *2011 18th IEEE International Conference on Image Processing*, 1541–1544. IEEE.

Rasch, B., and Born, J. 2013. About sleep's role in memory. *Physiol Rev* 93:681–766.

Redko, I.; Habrard, A.; and Sebban, M. 2017. Theoretical analysis of domain adaptation with optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 737–753. Springer.

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

Rostami, M.; Kolouri, S.; and Pilly, P. 2019. Complementary learning for overcoming catastrophic forgetting using experience replay. In *IJCAI*.

Ruvolo, P., and Eaton, E. 2013. Ella: An efficient lifelong learning algorithm. In *International Conference on Machine Learning*, 507–515.

Saxe, A. M.; McClelland, J. L.; and Ganguli, S. 2019. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences* 201820226.

Shalev-Shwartz, S., and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

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

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 4077–4087.