Optimal Continual Learning has Perfect Memory and is NP-HARD

Jeremias Knoblauch [1] **Hisham Husain** [2] **Tom Diethe** [3]

Abstract

Continual Learning (CL) algorithms incrementally learn a predictor or representation across multiple sequentially observed tasks. Designing CL algorithms that perform reliably and avoid so-called *catastrophic forgetting* has proven a persistent challenge. The current paper develops a theoretical approach that explains why. In particular, we derive the computational properties which CL algorithms would have to possess in order to avoid catastrophic forgetting. Our main finding is that such *optimal* CL algorithms generally solve an NP-HARD problem and will require perfect memory to do so. The findings are of theoretical interest, but also explain the excellent performance of CL algorithms using experience replay, episodic memory and core sets relative to regularization-based approaches.

1. Introduction

Continual Learning (CL) is a machine learning paradigm which takes inspiration from the ways in which biological organisms learn in the real world: Rather than observing a set of independent and identically distributed observations, CL seeks to design algorithms that sequentially learn from observations corresponding to different tasks. Unlike biological organisms, the artificial neural networks used for solving this problem suffer from *catastropic forgetting* (McCloskey & Cohen, 1989). Simply put, this phenomenon describes that sequentially learning on an increasing number of tasks will eventually yield increasingly poor representations of and predictions on previously observed tasks.

CL algorithms are not only a challenging research topic, they are also of tremendous practical importance (see e.g. Diethe et al., 2018): Often, it is impractical or even impossible to re-train a model every time new data arrives. For example, data may be too sensitive or expensive to store long term. Even if the storage of data is not a problem, increasingly complex models may make re-training computationally prohibitive.

While algorithms tackling the issue have steadily improved over the last few years, the CL problem has remained a persistently difficult challenge. In response, a growing body of research has designed novel CL algorithms based on different paradigms. In an attempt to structure the research output of the field, several paper have sought to classify these different paradigms (Parisi et al., 2019; Farquhar & Gal, 2018; van de Ven & Tolias, 2019). Broadly speaking, one can divide existing CL algorithms into one of three families: Regularization-based approaches (e.g. Kirkpatrick et al., 2017; Zenke et al., 2017; Schwarz et al., 2018; Ritter et al., 2018; Chaudhry et al., 2018; Lomonaco et al., 2019), replay-based approaches (Shin et al., 2017; Lopez-Paz & Ranzato, 2017; Kamra et al., 2017; Rolnick et al., 2019) as well as Bayesian and variationally Bayesian approaches (e.g. Nguyen et al., 2017; Tseran et al., 2018; Moreno-Munoz et al., 2019; Titsias et al., 2019). Other successful methods include learning a set of new parameters per task without discarding previously learnt ones (e.g. Rusu et al., 2016) as well as methods inspired by nearest-neighbour type considerations (e.g. Rebuffi et al., 2017).

For the purposes of this paper, it is important to contrast the CL setting with more traditional sequential learning paradigms. The idea of learning inputs sequentially is all but new—in fact, Lauritzen (1981) dates its origins as early as the 1880s. In clear contrast to the CL setting however, more traditional approaches rely on dependence assumptions between consecutive observations and tasks. For example, the work of Opper & Winther (1998) introduces an approximately Bayesian algorithm for neural networks that can be updated continuously in an on-line fashion. In clear opposition to the CL paradigm, the algorithm assumes that all observations were generated by the same and correctly specified data generating mechanism. Though restrictive, this also enables the authors to demonstrate asymptotic efficiency. Similarly, the contributions to sequential and streaming learning made by Honkela & Valpola (2003), Broderick et al. (2013) or McInerney et al. (2015) are geared towards streaming settings in which the differences in observations

[1]Department of Statistics, Warwick University & The Alan Turing Institute, London [2]Australian National University, Canberra & Data61, Sydney [3]Amazon. Correspondence to: Jeremias Knoblauch <j.knoblauchwarwick.ac.uk>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

vary only very slowly. Clearly, these methods are feasible only on a restricted range of problem settings. While this makes them less universally applicable, it also makes them reliable and their failures more interpretable. In clear distinction to this, CL algorithms seek to mimic biological brains and thus do not impose assumptions about the dependence between observations and tasks.

This has led to methods that can produce reasonable results over a wider range of settings, but also implies that CL algorithms generally have no clear performance guarantees. Worse still, it often leaves open in which contexts these algorithms can be expected to function reliably (see e.g. van de Ven & Tolias, 2019). Does this mean that it is generally computationally impossible to produce reliable CL algorithms without making assumptions on the task distributions? In this paper, we develop theoretical arguments which confirm this suspicion: Optimal CL algorithms would have to solve an NP-HARD problem and perfectly memorize the past.[1] This is not a purely negative result, however: The requirement of perfect memory in particular is of fundamental practical interest. Specifically, it explains recent results that favour approaches based on replay, episodic memory and core sets relative to regularization-based approaches (see e.g. Nguyen et al., 2017; van de Ven & Tolias, 2019). Thus, the findings also show that methods (approximately) recalling or reconstructing observations from previously observed tasks will be the most promising in developing reliable CL algorithms.

In this paper, we introduce a definition of CL wide enough to encompass the large number of competing approaches introduced over the last few years. Further, we define an equally flexible notion of *optimality* for CL algorithms. In the context of this paper, these optimality criteria are used to rigorously define what catastrophic forgetting entails: A CL algorithm which is optimal with respect to a given criterion avoids catastrophic forgetting (as formalized by this criterion). We then ask a central question:

> *What are the computational properties of an optimal* CL *algorithm?*

In answering this question, we develop new insights into the design requirements for CL algorithms that avoid catastrophic forgetting and provide the first thoroughly theoretical treatment of CL algorithms:

(1) We show that without any further assumptions, the optimality of CL algorithms can be studied with the tools of set theory (Lemma 1), which drastically simplifies the subsequent analysis.

(2) We show that optimal CL algorithms can solve a version of the set intersection decision problem (Lemma 2). Crucially, this decision problem will generally be NP-COMPLETE, meaning that the optimal CL algorithm itself is NP-HARD (Theorem 1; Corollary 1).

(3) We define the notion of an equivalence sets and use their properties (Lemma 3) to motivate the definition of perfect memory: Specifically, we say that a CL algorithm has perfect memory if it stores at least one element from each equivalence set (Definition 11).

(4) Re-using the decision problem of Lemma 2, we show that optimal CL has perfect memory under mild regularity conditions (Theorem 2; Corollary 2).

Our findings illuminate that CL algorithms can be seen as polynomial time heuristics targeted at solving an NP-HARD problem. Further, they explain why mimicking the perfect memory requirement of optimal CL through memorization heuristics generally outperforms CL algorithms based on regularization heuristics. Throughout, we make an effort to provide proof sketches for all of the most important findings. Detailed derivations for all claims can be found in the Appendix.

The remainder of this paper is structured as follows: In Section 2, we introduce basic notation and concepts. Next, we define CL algorithms and their optimality in Section 3. We then show that any optimal CL algorithm can be expressed in an idealized alternative form in Section 4. This idealized form is interesting because it drastically simplifies the analysis. Fourth, we show that an optimal CL algorithm can solve a set intersection decision problem. Under mild conditions, this decision problem is NP-COMPLETE, which we use to prove that the corresponding optimization problem (i.e., optimal CL) is NP-HARD in Section 5.1. Fifth, in Section 5.2 we define a notion of perfect memory that is suitable for CL algorithms. We then demonstrate that optimal CL algorithms will generally have perfect memory. Lastly, Section 6 provides a brief discussion of the implications of our results for CL algorithm design.[2]

2. Preliminaries

Throughout, we deal with random variables $\boldsymbol{X}_t, \boldsymbol{Y}_t$. Realizations of the random variable $\boldsymbol{X}_t$ live on the input space $\mathcal{X}$ and provide information about the random outputs $\boldsymbol{Y}_t$ with realizations on the output space $\mathcal{Y}$. Throughout, $\mathcal{P}(A)$ denotes the collection of all probability measures on A.

[1] Unlike other results (e.g. Blum & Rivest, 1992), these findings are not specific to using Neural networks or any other type of specific learning algorithm. Rather, they are borne out of to the very nature of the CL setting. To illustrate this point, we give an NP-HARD linear regression example throughout.

[2] A video summary of these contributions is available at https://www.youtube.com/watch?v=Ma-H37QtkM8

Definition 1 (Tasks)**.** *For a number $T \in \mathbb{N}$ and random variables $\{(\boldsymbol{X}_t, \boldsymbol{Y}_t)\}_{t=1}^T$ defined on the same spaces $\mathcal{X}$ and $\mathcal{Y}$, the random variable $(\boldsymbol{X}_t, \boldsymbol{Y}_t)$ is the t-th task, and its probability space is $(\mathcal{X}_t \times \mathcal{Y}_t, \boldsymbol{\Sigma}, \mathbb{P}_t)$, where $\boldsymbol{\Sigma}$ is a σ-algebra and $\mathbb{P}_t$ a probability measure on $\mathcal{X}_t \times \mathcal{Y}_t \subseteq \mathcal{X} \times \mathcal{Y}$.*

Given a sequence of samples from task-specific random variables, a CL algorithm sequentially learns a predictor for $\boldsymbol{Y}_t$ given $\boldsymbol{X}_t$. This means that there will be some hypothesis class $\mathcal{F}_{\boldsymbol{\Theta}}$ consisting of conditional distributions which allow (probabilistic) predictions about likely values of $\boldsymbol{Y}_t$.

Definition 2 (CL hypothesis class)**.** *The* CL *hypothesis class $\mathcal{F}_{\boldsymbol{\Theta}}$ is parameterized by $\boldsymbol{\Theta}$: For any $f \in \mathcal{F}_{\boldsymbol{\Theta}}$, there exists a $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ so that $f_{\boldsymbol{\theta}} = f$. More precisely, $\mathcal{F}_{\boldsymbol{\Theta}} \subset \mathcal{P}(\mathcal{Y})^{\mathcal{X}}$ if the task label is not conditioned on. Alternatively, $\mathcal{F}_{\boldsymbol{\Theta}} \subset \mathcal{P}(\mathcal{Y})^{\mathcal{X} \times \{1,2,\dots T\}}$ if the label is conditioned on.*

Remark 1. *Note that while this formulation may seem to exclude Bayesian approaches at first glance, this is not the case. In fact, one simply notes that a posterior distribution acts as an (infinite-dimensional) parameter: Specifically, suppose we have some model $m_{\boldsymbol{\kappa}}$ parameterized by a finite-dimensional parameter $\boldsymbol{\kappa} \in \boldsymbol{K}$ and want to form a posterior belief about it. In this case, we could recover the Bayesian approach by setting $\boldsymbol{\Theta} = \mathcal{P}(\boldsymbol{K})$ to be the collection of possible posteriors and $f_{\boldsymbol{\theta}}(x) = \int_{\boldsymbol{K}} m_{\boldsymbol{\kappa}}(x) d\boldsymbol{\theta}(\boldsymbol{\kappa})$.*

Remark 2. *The set-valued indicator functions $1_y = 1_{\{y\}}$ are elements of $\mathcal{P}(\mathcal{Y})$ for all $y \in \mathcal{Y}$. Thus, $\mathcal{Y}^{\mathcal{X}} \subset \mathcal{P}(\mathcal{Y})^{\mathcal{X}}$ and $\mathcal{Y}^{\mathcal{X} \times \{1,2,\dots T\}} \subset \mathcal{P}(\mathcal{Y})^{\mathcal{X} \times \{1,2,\dots T\}}$. In other words, defining the hypothesis class as conditional distributions also recovers deterministic input-output mappings (via degenerate conditional distributions). For example, one may choose $h \in \mathcal{Y}^{\mathcal{X}}$ (or $h \in \mathcal{Y}^{\mathcal{X} \times \{1,2,\dots T\}}$) and construct $p(\boldsymbol{Y}|x) = 1_{h(x)=y} \cdot y$ (or $p(\boldsymbol{Y}|x,t) = 1_{h(x,t)=y} \cdot y$).*

Remark 3. *All results derived and definitions provided in the remainder can be modified in obvious ways to account for the case where $\mathcal{F}_{\boldsymbol{\Theta}} \subseteq \mathcal{P}(\mathcal{Y})^{\mathcal{X} \times \{1,2,\dots T\}}$. To keep notation as simple as possible however, we will assume that $\mathcal{F}_{\boldsymbol{\Theta}} \subseteq \mathcal{P}(\mathcal{Y})^{\mathcal{X}}$.*

3. Continual Learning & Optimality

Having defined both tasks and hypothesis classes, we now define the collection of procedures that constitute CL algorithms. To the best of our knowledge, this definition is wide enough to encompass any existing CL algorithm. Figure 1 visualizes this definition.

Definition 3 (Continual Learning)**.** *For a* CL *hypothesis class $\mathcal{F}_{\boldsymbol{\Theta}}$, $T \in \mathbb{N}$ and any sequence of probability measures $\{\widehat{\mathbb{P}}_t\}_{t=1}^T$ such that $\widehat{\mathbb{P}}_t \in \mathcal{P}(\mathcal{Y}_t)^{\mathcal{X}_t} \subseteq \mathcal{P}(\mathcal{Y})^{\mathcal{X}}$,* CL *algorithms are specified by functions*

$$\widehat{\mathcal{A}}_{\mathrm{I}} : \boldsymbol{\Theta} \times \mathcal{I} \times \mathcal{P}(\mathcal{Y})^{\mathcal{X}} \to \mathcal{I}$$
$$\widehat{\mathcal{A}}_{\boldsymbol{\theta}} : \boldsymbol{\Theta} \times \mathcal{I} \times \mathcal{P}(\mathcal{Y})^{\mathcal{X}} \to \boldsymbol{\Theta},$$

where $\mathcal{I}$ is some space that may vary between different CL *algorithms. Given $\mathcal{A}_{\mathrm{I}}$ and $\mathcal{A}_{\boldsymbol{\theta}}$ and some initializations $\boldsymbol{\theta}_0$ and I_0,* CL *defines a procedure given by*

$$\begin{aligned}
\boldsymbol{\theta}_1 &= \widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_0, \mathrm{I}_0, \widehat{\mathbb{P}}_1) \\
\mathrm{I}_1 &= \widehat{\mathcal{A}}_{\mathrm{I}}(\boldsymbol{\theta}_1, \mathrm{I}_0, \widehat{\mathbb{P}}_1) \\
\boldsymbol{\theta}_2 &= \widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_1, \mathrm{I}_1, \widehat{\mathbb{P}}_2) \\
\mathrm{I}_2 &= \widehat{\mathcal{A}}_{\mathrm{I}}(\boldsymbol{\theta}_2, \mathrm{I}_1, \widehat{\mathbb{P}}_2) \\
&\dots \\
\boldsymbol{\theta}_T &= \widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{T-1}, \mathrm{I}_{T-1}, \widehat{\mathbb{P}}_T) \\
\mathrm{I}_T &= \widehat{\mathcal{A}}_{\mathrm{I}}(\boldsymbol{\theta}_T, \mathrm{I}_{T-1}, \widehat{\mathbb{P}}_T).
\end{aligned}$$

Remark 4. *In practice, the probability measures $\widehat{\mathbb{P}}_t$ will be empirical measures of $(\boldsymbol{X}_t, \boldsymbol{Y}_t)$ that are constructed from a finite number of samples.*

Remark 5. *An extremely attractive feature of the above definition is its generality. In partiular, the quantities $\mathrm{I}_t \in \mathcal{I}$ are interpretable as any kind of additional information carried forward through time. While the role of these objects will differ between* CL *algorithms, our definition is suitable to describe all of them. For example, in elastic weight consolidation (Kirkpatrick et al., 2017; Ritter et al., 2018), I_t will be a diagonalized approximation of Fisher information. In contrast, variational continual learning (Nguyen et al., 2017) will store the approximate posterior on the previous $t-1$ tasks as well as core sets for the previous tasks in I_t. Generally, I_t dictates the memory requirements of any* CL *algorithm. While the memory requirement is usually constant in the number of tasks, the* CL *algorithm in Rusu et al. (2016) would induce linearly growing memory requirements, as it carries all previously fitted parameters $\{\boldsymbol{\theta}_i\}_{i=1}^t$ forward in time.*

Remark 6. *Throughout the paper, whenever we write $\boldsymbol{\theta}_t$, this value should be understood as a recursively defined function of all previously observed tasks $\{\widehat{\mathbb{P}}_i\}_{i=1}^t$:*

$$\begin{aligned}
\boldsymbol{\theta}_t &= \widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{t-1}, \boldsymbol{I}_{t-1}, \widehat{\mathbb{P}}_t) \\
&= \widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{t-2}, \boldsymbol{I}_{t-2}, \widehat{\mathbb{P}}_{t-1}), \boldsymbol{I}_{t-1}, \widehat{\mathbb{P}}_t) \\
&= \widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{t-2}, \boldsymbol{I}_{t-2}, \widehat{\mathbb{P}}_{t-1}), \widehat{\mathcal{A}}_{\mathrm{I}}(\boldsymbol{\theta}_{t-2}, \boldsymbol{I}_{t-2}, \widehat{\mathbb{P}}_{t-1}), \widehat{\mathbb{P}}_t) \\
&= \dots
\end{aligned}$$

Clearly, a similar logic applies to the information I_t passed forward through time. Put differently, whenever we write $\boldsymbol{\theta}_t$ and I_t throughout this paper, it is instructive to think about them as functions evaluated at all previous tasks, i.e.

$$\boldsymbol{\theta}_t = \mathcal{B}_{\boldsymbol{\theta}}\left(\{\widehat{\mathbb{P}}_i\}_{i=1}^t\right)$$
$$\mathrm{I}_t = \mathcal{B}_{\mathrm{I}}\left(\{\widehat{\mathbb{P}}_i\}_{i=1}^t\right),$$

for functions $\mathcal{B}_{\boldsymbol{\theta}}, \mathcal{B}_{\mathrm{I}}$ specified implicitly via $\widehat{\mathcal{A}}_{\boldsymbol{\theta}}$ and $\widehat{\mathcal{A}}_{\boldsymbol{I}_t}$.

While the literature on CL has studied the problem of *catastrophic forgetting* empirically, to the best of our knowledge no previous theoretical study has been attempted. Thus, we first need to introduce a formal way of assessing whether a CL algorithm suffers catastrophic forgetting. As different researchers might disagree on the precise meaning of catastrophic forgetting, our formalism is very flexible. In particular, all that it needs is an arbitrary binary-valued optimality criterion $\mathcal{C}$, whose function is to assess whether or not information of a task has been retained ($\mathcal{C} = 1$) or forgotten ($\mathcal{C} = 0$). According to this formalism, a CL algorithm avoids catastrophic forgetting (as judged by the criterion $\mathcal{C}$) if and only if its output at task t is guaranteed to satisfy $\mathcal{C}$ on all previously seen tasks. In this context, different ideas about the meaning of catastrophic forgetting would result in different choices for $\mathcal{C}$. As we will analyze CL with the tools of set theory, it is also convenient to define the function SAT, which maps from task distributions into the subsets consisting of all values in $\boldsymbol{\Theta}$ which satisfy the criterion $\mathcal{C}$ on the given task.

Definition 4. *For an optimality criterion $\mathcal{C} : \boldsymbol{\Theta} \times \mathcal{P}(\mathcal{X} \times \mathcal{Y}) \to \{0, 1\}$ and a set $\mathcal{Q} \subseteq \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ of task distributions, the function $\text{SAT} : \mathcal{P}(\mathcal{X} \times \mathcal{Y}) \to 2^{\boldsymbol{\Theta}}$ defines the subset of $\boldsymbol{\Theta}$ which satisfies $\mathcal{C}$ and is given by*

$$\text{SAT}(\widehat{\mathbb{P}}) = \{\boldsymbol{\theta} \in \boldsymbol{\Theta} : \mathcal{C}(\boldsymbol{\theta}, \widehat{\mathbb{P}}) = 1\}.$$

The collection of all possible sets generated by SAT *is*

$$\text{SAT}_{\mathcal{Q}} = \{\text{SAT}(\widehat{\mathbb{P}}) : \widehat{\mathbb{P}} \in \mathcal{Q}\}$$

and the collection of finite intersections from $\text{SAT}_{\mathcal{Q}}$ is

$$\text{SAT}_{\cap} = \{\cap_{i=1}^{t} A_i : A_i \in \text{SAT}_{\mathcal{Q}}, \\ 1 \leq i \leq t \text{ and } 1 \leq t \leq T,\ T \in \mathbb{N}\}.$$

Lastly, for a given sequence $\{\widehat{\mathbb{P}}_t\}_{t=1}^{T}$ in $\mathcal{Q}$, define

$$\text{SAT}_t = \text{SAT}(\widehat{\mathbb{P}}_t)$$
$$\text{SAT}_{1:t} = \cap_{i=1}^{t} \text{SAT}_i,$$

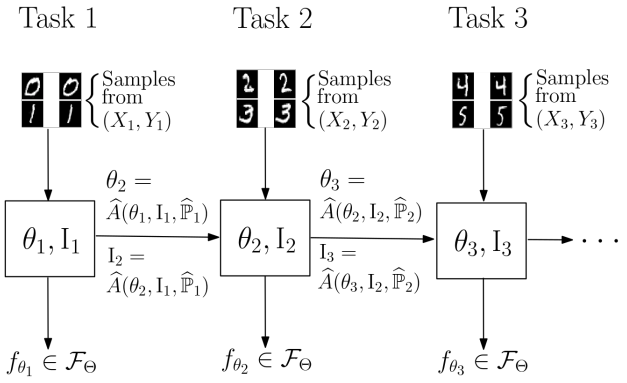

Figure 1. Schematic of a generic CL algorithm, using notation introduced in Section 2 and Definition 3.

for all $t = 1, 2, \ldots T$.

Definition 5 (Optimality)**.** *A* CL *algorithm is optimal with respect to the criterion $\mathcal{C}$ and a set $\mathcal{Q}$ of task distributions if*

(i) for any sequence $\{\widehat{\mathbb{P}}_t\}_{t=1}^{T}$ in $\mathcal{Q}$, $\mathcal{C}(\boldsymbol{\theta}_t, \widehat{\mathbb{P}}_i) = 1$, for all $i = 1, 2, \ldots t$ and all $t = 1, 2, \ldots T$;

(ii) it holds that for any fixed $\boldsymbol{\theta}', \mathrm{I}'$ that $\widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}', \mathrm{I}', \widehat{\mathbb{P}}) = \widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}', \mathrm{I}', \widehat{\mathbb{Q}})$ and $\widehat{\mathcal{A}}_{\mathrm{I}}(\boldsymbol{\theta}', \mathrm{I}', \widehat{\mathbb{P}}) = \widehat{\mathcal{A}}_{\mathrm{I}}(\boldsymbol{\theta}', \mathrm{I}', \widehat{\mathbb{Q}})$ if $\text{SAT}(\widehat{\mathbb{Q}}) = \text{SAT}(\widehat{\mathbb{P}})$, for all $\widehat{\mathbb{P}}, \widehat{\mathbb{Q}}$ in $\mathcal{Q}$,

for any $T \in \mathbb{N}$.

Remark 7. *Note that we have defined optimality with respect to a possibly restricted subclass $\mathcal{Q} \subseteq \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ of task distributions. While the bulk of the literature on* CL *makes no assumptions on the set of task distributions that the algorithm processes, this ensures that our notion of optimality could be made arbitrarily strict.*

Remark 8. *In spite of its name, the above definition only imposes weak restrictions on a* CL *algorithm to be called optimal: All that it requires is that some arbitrary criterion $\mathcal{C}$ is satisfied on each task.*

Remark 9. *While this is notationally suppressed, the criterion $\mathcal{C}$ itself could depend on some hyperparameter. For example, suppose that for some loss function $\ell : \boldsymbol{\Theta} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and some $\varepsilon \geq 0$, the optimality criterion is given by*

$$\mathcal{C}(\boldsymbol{\theta}, \widehat{\mathbb{P}}) = \begin{cases} 1 & \text{if } \int_{\mathcal{X} \times \mathcal{Y}} \ell(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) d\widehat{\mathbb{P}}(\boldsymbol{x}, \boldsymbol{y}) \leq \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$

Now $\mathcal{C} = \mathcal{C}_\varepsilon$ depends on ε so that one can define optimality for any fixed value of $\varepsilon \geq 0$.

4. A Convenient Idealization

Throughout, it will be convenient to derive results relative to a version of CL that is idealized. This idealization *directly* has access to the sets SAT_t (rather than to $\widehat{\mathbb{P}}_t$). In other words, the idealization has access to a convenient oracle: It is already informed of all elements of the hypothesis class $\mathcal{F}_{\boldsymbol{\Theta}}$ that satisfy the criterion $\mathcal{C}$ on the t-th task. Importantly and as we will show in Lemma 1, studying optimality with the idealized version of CL instead of the standard version does not impose any assumptions. As the idealized version of CL relies on basic set operations, this substantially simplifies the subsequent analysis.

Definition 6 (Idealized Continual Learning)**.** *For a hypothesis class $\mathcal{F}_{\boldsymbol{\Theta}}$, any $T \in \mathbb{N}$ and any sequence of sets $\{\text{SAT}_t\}_{t=1}^{T}$ generated with some fixed criterion $\mathcal{C}$ and an arbitrary sequence of probability measures $\{\widehat{\mathbb{P}}_t\}_{t=1}^{T}$ as in Definition 5, Idealized* CL *(Idealized* CL*) algorithms are specified by functions $\mathcal{A}_{\mathrm{I}}$ and $\mathcal{A}_{\boldsymbol{\theta}}$*

$$\mathcal{A}_{\mathrm{I}} : \boldsymbol{\Theta} \times \mathcal{I} \times \text{SAT}_{\mathcal{Q}} \to \mathcal{I}$$
$$\mathcal{A}_{\boldsymbol{\theta}} : \boldsymbol{\Theta} \times \mathcal{I} \times \text{SAT}_{\mathcal{Q}} \to \boldsymbol{\Theta},$$

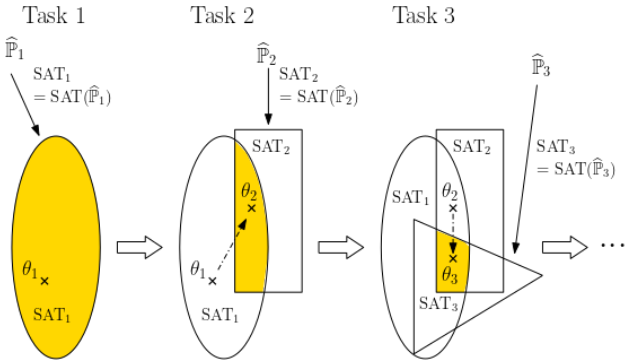

Figure 2. Visualization of optimal Idealized CL: At each task, the new value for $\boldsymbol{\theta}_t$ must lie in the intersection $\mathrm{SAT}_{1:t} = \cap_{i=1}^{t} \mathrm{SAT}_i$.

where $\mathcal{I}$ is some space that may vary between different Idealized CL *algorithms. Given $\mathcal{A}_\mathrm{I}$ and $\mathcal{A}_{\boldsymbol{\theta}}$ and some initializations $\boldsymbol{\theta}_0$ and I_0, Idealized* CL *defines a procedure given by*

$$
\begin{aligned}
\boldsymbol{\theta}_1 &= \mathcal{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_0, \mathrm{I}_0, \mathrm{SAT}_1) \\
\mathrm{I}_1 &= \mathcal{A}_\mathrm{I}(\boldsymbol{\theta}_1, \mathrm{I}_0, \mathrm{SAT}_1) \\
\boldsymbol{\theta}_2 &= \mathcal{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_1, \mathrm{I}_1, \mathrm{SAT}_2) \\
\mathrm{I}_2 &= \mathcal{A}_\mathrm{I}(\boldsymbol{\theta}_2, \mathrm{I}_1, \mathrm{SAT}_2) \\
&\dots \\
\boldsymbol{\theta}_T &= \mathcal{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{T-1}, \mathrm{I}_{T-1}, \mathrm{SAT}_T) \\
\mathrm{I}_T &= \mathcal{A}_\mathrm{I}(\boldsymbol{\theta}_T, \mathrm{I}_{T-1}, \mathrm{SAT}_T).
\end{aligned}
$$

It should be clear that the only difference between CL and Idealized CL is the third argument: Rather than using $\widehat{\mathbb{P}}_t$, Idealized CL algorithms use SAT_t. Apart from that, everything else remains the same: I_t is still interpretable as additional information and $\widehat{\mathbb{P}}_t$ as an empirical measure composed of samples from the t-th task.

Definition 7 (Optimal Idealized Continual Learning)**.** *An Idealized* CL *algorithm is an optimal Idealized* CL *procedure if $\boldsymbol{\theta}_t \in \mathrm{SAT}_{1:t}$ for all $t = 1, 2, \dots T$.*

In contrast to an optimal CL algorithm, an optimal Idealized CL algorithm has a clear set-theoretic interpretation that is set out in Figure 2: $\boldsymbol{\theta}_t$ needs to lie in the intersection of the sets that mark out the subspaces of $\boldsymbol{\Theta}$ on which $\mathcal{C}$ is satisfied relative to all t tasks observed thus far. Combined with the next result, this will serve to abstract and simplify the further analysis of optimal CL.

Lemma 1. *Any optimal* CL *algorithm is an optimal Idealized* CL *algorithm relative to the same criterion.*

Proof. Suppose that $\widehat{\mathcal{A}}_{\boldsymbol{\theta}}$ and $\widehat{\mathcal{A}}_\mathrm{I}$ define an optimal CL algorithm. Simply define $\mathcal{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_t, \mathrm{I}_t, \mathrm{SAT}(\widehat{\mathbb{P}}_t)) = \widehat{\mathcal{A}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_t, \mathrm{I}_t, \widehat{\mathbb{P}}_t)$ and similarly $\mathcal{A}_\mathrm{I}(\boldsymbol{\theta}_t, \mathrm{I}_t, \mathrm{SAT}(\widehat{\mathbb{P}}_t)) = \widehat{\mathcal{A}}_\mathrm{I}(\boldsymbol{\theta}_t, \mathrm{I}_t, \widehat{\mathbb{P}}_t)$ as the functions specifying the corresponding Idealized CL algorithm. By definition, $\mathrm{SAT}_t = \mathrm{SAT}(\widehat{\mathbb{P}}_t)$ so that the reverse is immediate, too. □

Lemma 1 plays a central role throughout the rest of the paper: Based on the stated equivalence, one can use idealized optimal CL algorithms to analyze standard optimal CL algorithms. This has two advantages: Firstly, it drastically simplifies the analysis by reducing it to basic set theory. Secondly, it provides new ways of forming intuitions about the computational properties of optimal CL algorithms.

5. Main Results

Next, we summarize the main results: Generally,

(1) optimal CL algorithms are NP-HARD and

(2) optimal CL algorithms require perfect memory.

While we sketch the most important proofs, full details and derivations are deferred to the Appendix. To clearly convey the most important insights, we additionally provide examples and illustrations.

Before proceeding, we state another key lemma that is invaluable for both main results. Its role is to lower bound both the memory requirement and computational hardness of optimal CL algorithms with that of a well-studied decision problem which is illustrated in Figure 3.

Lemma 2. *An optimal* CL *algorithm is computationally at least as hard as deciding whether $A \cap B = \emptyset$, for $A \in \mathrm{SAT}_\cap$ and $B \in \mathrm{SAT}_\mathcal{Q}$.*

Proof sketch. By virtue of Lemma 1, it suffices to show this for the corresponding optimal Idealized CL algorithm. Since $\boldsymbol{\theta}_t \in \mathrm{SAT}_{1:t}$, optimal Idealized CL solves a particular optimization problem: In particular, optimal Idealized CL finds a $\boldsymbol{\theta}_t \in A \cap B$, for some $A = \mathrm{SAT}_{1:(t-1)} \in \mathrm{SAT}_\cap$ and $B = \mathrm{SAT}_t \in \mathrm{SAT}_\mathcal{Q}$. Clearly, finding an element in $A \cap B$ is at least as hard as determining whether $A \cap B = \emptyset$. □

5.1. Computational Complexity

Finally, we are in a position to formally state our result on the computational hardness of optimal CL.

Theorem 1. *If $\mathcal{Q}$ and $\mathcal{C}$ are such that $\mathrm{SAT}_\mathcal{Q} \supseteq S$ or $\mathrm{SAT}_\cap \supseteq S$ so that S is the set of tropical hypersurfaces or the set of polytopes on $\boldsymbol{\Theta}$, then optimal* CL *is* NP-HARD*.*

Proof sketch. First, we use Lemma 2: Optimal CL can correctly decide if $A \cap B = \emptyset$, for all $A \in \mathrm{SAT}_\cap$ and

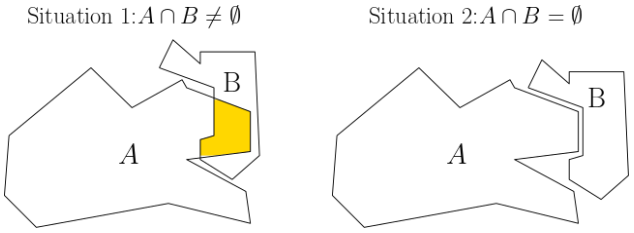

Figure 3. Any CL algorithm induces the two collections of sets $\text{SAT}_{\mathcal{Q}}$ and $\text{SAT}_{\cap}$. Lemma 2 says that if the CL algorithm is optimal, then it solves a problem at least as hard as deciding if $A \cap B = \emptyset$, for all $A \in \text{SAT}_{\cap}$ and $B \in \text{SAT}_{\mathcal{Q}}$.

$B \in \text{SAT}_{\mathcal{Q}}$. Second, we use established reductions to conclude that this decision problem is NP-COMPLETE. For the case where S is the set of tropical hypersurfaces, the results in Theobald (2006) can be used. If S is the set of polytopes, the same conclusion is reached by using the results of Tiwary (2008a) and Tiwary (2008b). Third, it then follows by standard arguments that the optimization problem corresponding to an NP-COMPLETE decision problem is NP-HARD. □

One may wonder how consequential the above result is in practice. Specifically, which kind of criterion $\mathcal{C}$ and which kind of model would produce polytopes or tropical hypersurfaces? In fact, relatively simple models and optimality criteria suffice to produce such adverse solution sets. We showcase this in the next example: As we shall see, a simple linear model together with an intuitively appealing upper bound on the prediction error as optimality criterion are sufficient to make the corresponding optimal CL problem NP-HARD.

Example 1. *Take $\mathcal{F}_{\boldsymbol{\Theta}}$ to be the collection of linear models with inputs on $\mathcal{X}$ and outputs on $\mathcal{Y} \subset \mathbb{R}$ linked through the coefficient vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Further, let $\mathcal{Q}$ be the collection of empirical measures*

$$\widehat{m}^t(y, x) = \frac{1}{n_t} \sum_{i=1}^{n_t} \delta_{(y_i^t, x_i^t)}(y, x)$$

whose $n_t \in \mathbb{N}$ atoms $\{(y_i^t, x_i^t)\}_{i=1}^{n_t}$ represent the t-th task. Further, define for $\varepsilon \geq 0$ and all $\mathbb{P} \in \mathcal{Q}$ the criterion

$$\mathcal{C}(\boldsymbol{\theta}, \widehat{\mathbb{P}}) = \begin{cases} 1 & \text{if } |y_i^t - \boldsymbol{\theta}^T x_i^t| \leq \varepsilon, \text{ for all } i = 1, 2, \ldots n_t \\ 0 & \text{otherwise.} \end{cases}$$

Then, it is straightforward to see that

$$\begin{aligned} \text{SAT}(\widehat{\mathbb{P}}) &= \left\{\boldsymbol{\theta} \in \boldsymbol{\Theta} : y_i^t - \boldsymbol{\theta}^T x_i^t \leq \varepsilon \text{ and } y_i^t - \boldsymbol{\theta}^T x_i^t - \geq \varepsilon, \right. \\ &\quad \left. \text{for all } i = 1, 2, \ldots n_t \right\} \\ &= \left(\cap_{i=1}^{n_t} \left\{\boldsymbol{\theta} \in \boldsymbol{\Theta} : y_i^t - \boldsymbol{\theta}^T x_i^t \leq \varepsilon\right\}\right) \cap \\ &\quad \left(\cap_{i=1}^{n_t} \left\{\boldsymbol{\theta} \in \boldsymbol{\Theta} : y_i^t - \boldsymbol{\theta}^T x_i^t \geq -\varepsilon\right\}\right), \end{aligned}$$

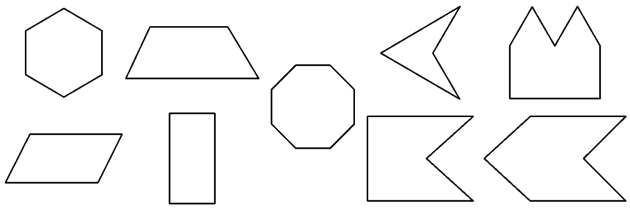

Figure 4. A small selection of polytopes in $\mathbb{R}^2$. Polytopes are basic and well-studied studied geometric shapes.

which is an intersection of $2n_t$ half-spaces in $\mathbb{R}^d$ and thus a polytope. Unless we make very strong assumptions about the common structure between the task distributions which generated the atoms, this implies that we could recover any given polytope in $\mathbb{R}^d$ by constructing $\text{SAT}(\widehat{\mathbb{P}})$. Under these circumstances, the conditions of Theorem 1 are met: $\text{SAT}_{\mathcal{Q}}$ contains all polytopes. While the criterion may seem strict, optimal CL *would still suffer the same problem even if we used the alternative criterion*

$$\mathcal{C}(\boldsymbol{\theta}, \widehat{\mathbb{P}}) = \begin{cases} 1 & \text{if } \frac{1}{n} \sum_{i=1}^{n} |y_i^t - \boldsymbol{\theta}^T x_i^t| \leq \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$

In this case, $\text{SAT}(\widehat{\mathbb{P}})$ would still be a polytope in $\mathbb{R}^d$, albeit made up only of $2d$ intersections of half-spaces. By similar reasoning as applied to $\text{SAT}_{\mathcal{Q}}$ before, the collection $\text{SAT}_{\cap}$ now contains any arbitrary polytope in $\mathbb{R}^d$.

Summarizing Example 1, the optimal CL problem is NP-HARD even for extremely simple models and even when we restrict the set of permissible data distributions to be almost comically simple. Indeed, the shapes depicted in Figure 4 are clearly far simpler than the sets of optimal solutions SAT_t that would be induced by non-linear hypothesis classes $\mathcal{F}_{\boldsymbol{\Theta}}$ such as Artificial Neural Networks. In other words, real world CL algorithms based on Deep Learning will induce a collection of sets $\text{SAT}_{\mathcal{Q}}$ whose elements are at least as hard to intersect as polytopes. Since even the intersection of the geometrically relatively simple polytopes is NP-HARD, it is straightforward to show that such CL algorithms solve an NP-HARD problem. The next Corollary formalizes this observation.

Corollary 1. *Suppose that $\text{SAT}_{\mathcal{Q}} \supseteq S$, with S being a collection of sets for which deciding if $A \cap B = \emptyset$ for $A, B \in S$ is computationally at least as hard as for the collection of polytopes in $\boldsymbol{\Theta}$. Then optimal* CL *is* NP-HARD.

5.2. Memory Requirements

Having established the computational hardness of optimal CL, we next investigate its memory requirements. To this end, we first need to develop a notion of perfect memory. Specifically, we will define perfect memory to be the most

memory-efficient way of retaining all solutions that have not been ruled out by previously processed tasks. The first step along this path is the definition of equivalence sets. Intuitively speaking, equivalence sets are subsets of $\mathbf{\Theta}$ whose values perform exactly the same across all tasks in $\mathcal{Q}$ (as judged by the criterion $\mathcal{C}$).

Definition 8 (Equivalence set)**.** *For $\boldsymbol{\theta} \in \mathbf{\Theta}$, define $S(\boldsymbol{\theta}) = \{A \in \text{SAT}_{\mathcal{Q}} : \boldsymbol{\theta} \in A\}$ and the equivalence sets*

$$\text{E}(\boldsymbol{\theta}) = \bigcap_{A \in S(\boldsymbol{\theta})} A.$$

Remark 10. *Equivalence sets are constructed as illustrated in Figure 5. Thus, any set $\text{E}(\boldsymbol{\theta})$ contains all other solutions $\boldsymbol{\theta}' \in \mathbf{\Theta}$ which are as good as $\boldsymbol{\theta}$. To illustrate this logic, suppose $\text{E}(\boldsymbol{\theta}) = \{\boldsymbol{\theta}\}$. In this case, for each value of $\boldsymbol{\theta} \in \mathbf{\Theta}$ there is **no** other value $\boldsymbol{\theta}' \in \mathbf{\Theta}$ that is guaranteed to perform equally well as $\boldsymbol{\theta}$ across all tasks in $\mathcal{Q}$.*

Equivalence sets satisfy a number of important properties summarized in the following Lemma.

Lemma 3. *For arbitrary equivalence sets $\text{E}(\boldsymbol{\theta}), \text{E}(\boldsymbol{\theta}')$ and arbitrary $A \in \text{SAT}_{\mathcal{Q}}$, it holds that*

- $\boldsymbol{\theta}' \in \text{E}(\boldsymbol{\theta}) \iff \text{E}(\boldsymbol{\theta}) = \text{E}(\boldsymbol{\theta}')$;
- *If $\boldsymbol{\theta}' \notin \text{E}(\boldsymbol{\theta})$, then $\text{E}(\boldsymbol{\theta}) \cap \text{E}(\boldsymbol{\theta}') = \emptyset$;*
- *Either $\text{E}(\boldsymbol{\theta}) \subseteq A$ or $\text{E}(\boldsymbol{\theta}) \cap A = \emptyset$.*

Next, we formally define perfect memory in the context of CL algorithms. In particular, we will say that a CL algorithm has perfect memory if it can reconstruct at least one element of each equivalence set whose elements satisfy the optimality criterion $\mathcal{C}$ on all tasks observed thus far. To this end, we define Minimal Covers and Minimal Representations. Figure 6 illustrates both concepts.

Definition 9 (Minimal cover)**.** *Given an indexed set $\{\boldsymbol{\theta}_i\}_{i \in I}$ of points in $\mathbf{\Theta}$, suppose that $\{\text{E}(\boldsymbol{\theta}_i)\}_{i \in I}$ forms a cover of $\cup_{A \in \text{SAT}_{\mathcal{Q}}} A$ such that for $i \neq j$, $\text{E}(\boldsymbol{\theta}_i) \cap \text{E}(\boldsymbol{\theta}_j) = \emptyset$. Then we call such a cover minimal.*

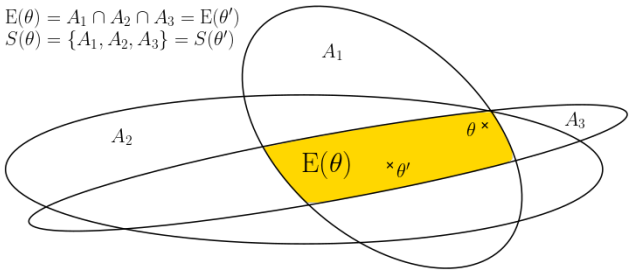

Figure 5. Illustration of Equivalence sets and the results of Lemma 3: $\text{E}(\boldsymbol{\theta})$ is the intersection of $S(\boldsymbol{\theta})$, the collection of all sets in $\text{SAT}_{\mathcal{Q}}$ that contain $\boldsymbol{\theta}$.

Remark 11. *Note that by virtue of Lemma 3, any Minimal cover forms a non-overlapping and unique partition of $\cup_{A \in \text{SAT}_{\mathcal{Q}}} A$. Figure 6 illustrates this point.*

Definition 10 (Minimal representation)**.** *Let $\{\text{E}(\boldsymbol{\theta}_i)\}_{i \in I}$ be a minimal cover. Denoting by $f : I \to \mathbf{\Theta}$ a function such that $f(i) \in \text{E}(\boldsymbol{\theta}_i)$ for all $i \in I$, we call the set $\{f(i)\}_{i \in I}$ a minimal representation of $\cup_{A \in \text{SAT}_{\mathcal{Q}}} A$.*

In the context of a CL algorithm, the minimal representation is the smallest possible set in $\mathbf{\Theta}$ that one needs to to retain all potential solutions of different quality (as judged by the criterion $\mathcal{C}$). In fact, it is instructive to think of a minimal representation as the most memory-efficient representation of the set of all potential solutions: Since all points in an equivalence set are equally good under $\mathcal{C}$ by definition, one can store a single point for each equivalence class $\text{E}(\boldsymbol{\theta}) \subset \mathbf{\Theta}$ without losing information. In other words, a minimal representation is the most memory-efficient way of retaining perfect memory.

Definition 11 (Perfect memory)**.** *We say that an optimal* CL *algorithm has perfect memory if there exists a function $h : \mathbf{\Theta} \times \mathcal{I} \to 2^{\mathbf{\Theta}}$ for which $h(\boldsymbol{\theta}_t, \text{I}_t) = C_t$ such that $\text{SAT}_{1:t} \supseteq C_t \supseteq (\cup_{i \in I} f(i)) \cap \text{SAT}_{1:t}$ at task $(t+1)$, for some fixed but arbitrary minimal representation $\{f(i)\}_{i \in I}$ and for any arbitrary $\text{SAT}_{1:t} \in \text{SAT}_{\cap}$.*

Remark 12. *The above conceptualizes an intuitive notion of perfect memory: The set $(\cup_{i \in I} f(i)) \cap \text{SAT}_{1:t}$ contains exactly one value for each equivalence set whose solutions are still optimal given the first t tasks. Thus, an optimal* CL *algorithm has perfect memory if it can reconstruct at least one value for each equivalence set that has not been ruled out as sub-optimal by the t preceding tasks. Equivalently, one could say that an optimal* CL *algorithm can perfectly memorize all equivalence sets ruled out by the first t tasks.*

We are now almost in a position to show that optimal CL algorithms will generally have perfect memory. The last missing ingredient is the following lemma.

Lemma 4. *If a* CL *algorithm is optimal, there exists h :*

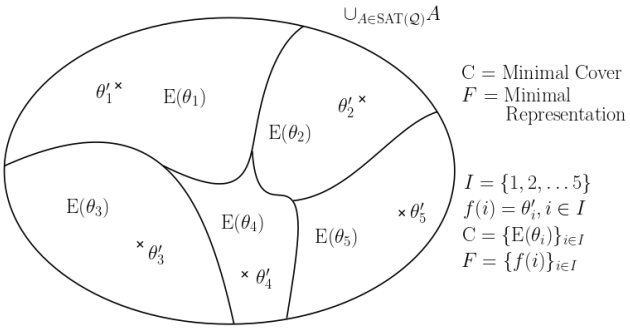

Figure 6. Illustration of Minimal Covers and Representations. Unlike Minimal Covers, Minimal Representations are *not* unique: Both $f(i) = \boldsymbol{\theta}_i$ and $f(i) = \boldsymbol{\theta}'_i$ yield Minimal Representations.

$\boldsymbol{\Theta} \times \mathcal{I} \to 2^{\boldsymbol{\Theta}}$ *for which* $h(\boldsymbol{\theta}_t, \mathrm{I}_t) = C_t$ *is such that* $C_t \cap A = \emptyset \iff \mathrm{SAT}_{1:t} \cap A = \emptyset$*, for all* $A \in \mathrm{SAT}_{\mathcal{Q}}$.

With this, all that is left to do is proving that C_t of Lemma 4 is contained by $\mathrm{SAT}_{1:t}$ and contains $(\cup_{i \in I} f(i)) \cap \mathrm{SAT}_{1:t}$, for some Minimal Representation $\{f(i)\}_{i \in I}$. Under mild regularity conditions, this yields the second main result.

Theorem 2. *Suppose that for an optimal* CL *algorithm,* $C_t \subseteq \mathrm{SAT}_{1:t}$*,* $C_t \subseteq C_{t-1}$ *and that for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ *there exists* $\{A_t\}_{t=1}^T$ *in* $\mathrm{SAT}_{\mathcal{Q}}$ *such that* $\cap_{t=1}^T A_t = \mathrm{E}(\boldsymbol{\theta})$*. Then this optimal* CL *algorithm has perfect memory.*

Proof sketch. By Lemma 4, we know that C_t will suffice to solve the decision problem already discussed in the proof sketch of Theorem 1. We also know that any optimal CL algorithm has to be able to solve this decision problem. Thus, the memory requirements of the decision problem lower bound those of optimal CL algorithms. Next, we prove that there must exist a minimal representation $\{f(i)\}_{i \in I}$ for which $C_t \supseteq (\cup_{i \in I} \{f(i)\}) \cap \mathrm{SAT}_{1:t}$, for all $t = 1, 2, \dots T$. We do this by combining Lemma 3 with the additional conditions imposed upon C_t. □

The conditions imposed in Theorem 2 are general, but also rather abstract. Alternatively, much stronger conditions with a more straightforward interpretation could be imposed to derive the same result.

Corollary 2. *Suppose* $\mathcal{C}$ *and* $\mathcal{Q}$ *are such that* $\mathrm{E}(\boldsymbol{\theta}) \in \mathrm{SAT}_{\mathcal{Q}}$ *for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$*. Then any optimal* CL *algorithm has perfect memory.*

Even though the conditions of Corollary 2 are far more restrictive than those of Theorem 2, it is relatively easy to find examples on which they hold.

Example 2. *To keep things simple, consider again the set-up of Example 1. Consider*

$$\varepsilon_t = \min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} |y_i^t - \boldsymbol{\theta}^T x_i^t|,$$

$$\boldsymbol{\theta}_t^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} |y_i^t - \boldsymbol{\theta}^T x_i^t|.$$

Unless we substantially restrict the permitted task distributions in $\mathcal{Q}$*, we cannot exclude the possibility that* $\mathrm{E}(\boldsymbol{\theta}) = \{\boldsymbol{\theta}\} \in \mathrm{SAT}_{\mathcal{Q}}$ *for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$*. Specifically, this is the case if for a fixed* $\varepsilon \geq 0$ *and for any* $\boldsymbol{\theta} \in \cup_{A \in \mathrm{SAT}_{\mathcal{Q}}} A$*, it is possible to find an empirical measure in* $\mathcal{Q}$ *constructed with atoms* $\{(y_i^t, x_i^t)\}_{i=1}^{n_t}$ *for which* $\boldsymbol{\theta} = \boldsymbol{\theta}_t^*$ *so that*

$$\frac{1}{n} \sum_{i=1}^{n} |y_i^t - \boldsymbol{\theta}^T x_i^t| = \varepsilon = \varepsilon_t = \frac{1}{n} \sum_{i=1}^{n} |y_i^t - (\boldsymbol{\theta}_t^*)^T x_i^t|.$$

For the corresponding empirical distribution $\widehat{\mathbb{P}}_t$*, it would then follow that* $\mathrm{SAT}(\widehat{\mathbb{P}}_t) = \{\boldsymbol{\theta}\}$*. Since* $\boldsymbol{\theta}$ *was chosen arbitrarily, this immediately entails that*

$$\mathrm{E}(\boldsymbol{\theta}) = \{\boldsymbol{\theta}\} \in \mathrm{SAT}_{\mathcal{Q}}$$

for all $\boldsymbol{\theta} \in \cup_{A \in \mathrm{SAT}_{\mathcal{Q}}} A$ *so that the conditions of Corollary 2 are satisfied. Notice that one could apply the same logic with most other predictors* $f_{\boldsymbol{\theta}}$ *by replacing* $\boldsymbol{\theta}^T x_i^t$ *with* $f_{\boldsymbol{\theta}}(x_i^t)$ *in the definition of* $\mathcal{C}(\boldsymbol{\theta}, \widehat{\mathbb{P}})$ *of Example 1.*

The take-away message from the previous example is that even though the conditions of Theorem 2 (or Corollary 2) will be harder to verify for more complicated model classes, they should be expected to hold in practice unless $\mathcal{Q}$ is substantially restricted and $\mathcal{C}$ is picked very carefully.

6. Implications for CL in the Wild

Our results are of theoretical interest, but also have two practical implications: Firstly, they illuminate that CL algorithms should be seen as polynomial time heuristics targeted at solving an NP-HARD problem. This new perspective explains why the design of reliable CL algorithms has proven a persistent challenge. Secondly, our results provide a theoretically grounded confirmation of recent benchmarking results, which found that CL algorithms based on experience replay, core sets and episodic memory were more reliable than regularization-based alternatives. In the remainder of this section, we elaborate on both of these points.

6.1. CL as Polynomial Time Heuristics

As we have shown, CL algorithms that avoid catastrophic forgetting as judged by an optimality criterion $\mathcal{C}$ generally solve NP-HARD problems. Consequently, the polynomial time heuristics that have been proposed to (sub-optimally) tackle the CL problem in practice can be seen as heuristic algorithms without performance guarantees. As these heuristics are not coupled to explicit assumptions on the data generating mechanisms underlying the tasks, it is easy to see why reliable CL algorithms have proven to be a persistent challenge. Since many well-known NP-HARD problems admit heuristic polynomial time approximation algorithms *with* performance guarantees, this also raises the question whether one could derive such algorithms for CL. So far however, this has not been attempted. Instead, the literature has focused on two useful heuristics for the design of CL algorithms: Memorization and regularization approaches.

6.2. Memorization versus Regularization

In the recent large-scale comparative study of van de Ven & Tolias (2019), replay- and memorization-based CL heuristics were found to produce far more reliable results than regularization-based approaches. Similarly, Nguyen et al.

(2017) found that a variant of their (approximately Bayesian) algorithm which used core sets to represent previously seen tasks produced substantially improved results over the version without core sets. In the same vein, Farquhar & Gal (2018) found that using generative approaches to complement approximately Bayesian procedures improved performance. Most recently, Titsias et al. (2019) outperformed competing approaches using inducing points within the Gaussian Process framework as efficient summaries of previous observations.

In fact, these empirical finding are to be expected given the perfect memory requirement of optimal CL: Approaches based on replay and core sets amount to storing an approximate representation of previous tasks. In other words, these CL algorithms store information I_t such that one can reconstruct an approximation $\mathbb{Q}_{1:t} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ for the joint distribution over all tasks observed thus far. In this sense, it is instructive to think of them as processing a single (albeit consecutively modified) task $\mathbb{Q}_{1:t}$. Intuitively then, CL algorithms of this kind will perform well under two conditions: It must be relatively easy to find some element $\boldsymbol{\theta} \in A$ for a *single* $A \in \mathrm{SAT}_{\mathcal{Q}}$ and it must hold that $\mathrm{SAT}(\mathbb{Q}_{1:t}) \approx \mathrm{SAT}_{1:t}$. The next Example and Figure 7 expand on this point.

Example 3. *Keeping things simple, we construct $\mathcal{Q}$ and $\mathcal{C}$ to ensure that $\mathrm{SAT}_{\mathcal{Q}}$ consists only of spheres. Suppose that $\boldsymbol{\theta} \in \boldsymbol{\Theta} = \mathbb{R}^2$ represents the (two-dimensional) mean across all tasks and that $\mathcal{Q}$ consists of empirical measures. Further, suppose that $\mathcal{X} = \emptyset$ (so that there only is an output variable $\boldsymbol{Y}_t$) and that the criterion of interest is an upper bound of ε on the average mean squared Euclidean distance, i.e.*

$$\mathcal{C}(\boldsymbol{\theta}, \widehat{\mathbb{P}}) = \begin{cases} 1 & \text{if } \frac{1}{n}\sum_{i=1}^{n} \|y_i^t - \boldsymbol{\theta}\|_2^2 \leq \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$

For a single task it is easy to find a value of $\boldsymbol{\theta}$ satisfying $\mathcal{C}$ by using simple linear regression (provided that ε is chosen large enough). Figure 7 illustrates why CL *based on core sets, replay or memory can typically be expected to work relatively well.*

In contrast to memorization-based heuristics, regularization-based approaches have to make inappropriate assumptions about the difficulty of moving from $\mathrm{SAT}_{1:(t-1)}$ to $\mathrm{SAT}_{1:t}$. Specifically, the choice of regularizer corresponds to an implicit and strong assumption on the geometry and nature of overlapping regions between $\mathrm{SAT}_{1:(t-1)}$ and SAT_t. As this implicit assumption is usually severely violated in practice, regularization-based CL algorithms often underperform, especially when the number of tasks is very large (van de Ven & Tolias, 2019; Lomonaco et al., 2019).

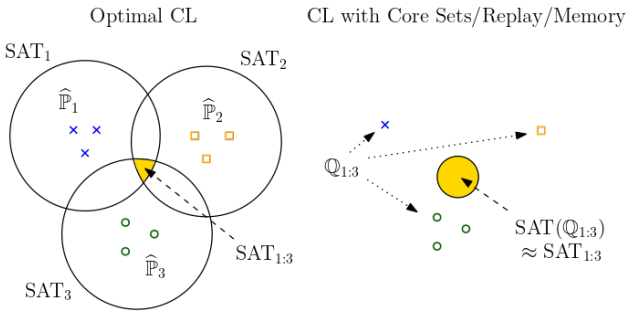

Figure 7. **Left:** Optimal CL finds an element $\boldsymbol{\theta}_t \in \mathrm{SAT}_{1:t}$. **Right:** CL algorithms based on core sets/replay/memory find an element $\boldsymbol{\theta}_t \in \mathrm{SAT}(\mathbb{Q}_{1:t})$, which typically is sufficiently similar to $\mathrm{SAT}_{1:t}$.

7. Conclusion

With this paper, we have produced the first generic theoretical study of the Continual Learning (CL) problem. We did so by translating the notion of catastrophic forgetting into the language of basic set theory. With this in hand, we showed that optimal CL is generally NP-HARD and requires perfect memory of the past. This has two practical ramifications: Firstly, it illustrates that existing CL algorithms can be seen as polynomial time heuristics targeted at solving an NP-HARD problem. Secondly, it reveals why memorization-based CL approaches using experience replay, core sets or episodic memory have generally proven more successful than their regularization-based alternatives.

Acknowledgements

We thank Andreas Damianou and Shuai Tang for fruitful discussions. Moreover, we thank Isak Falk, Juan Maroñas, and Ollie Hammelijnck for spotting a number of typos in the manuscript and Hans Kersting for the gym tour.

JK is funded by EPSRC grant EP/L016710/1 as part of the Oxford-Warwick Statistics Programme (OxWaSP) as well as by the Facebook Fellowship Programme. JK is also supported by the Lloyds Register Foundation programme on Data Centric Engineering through the London Air Quality project and by The Alan Turing Institute for Data Science and AI under EPSRC grant EP/N510129/1 in collaboration with the Greater London Authority.

References

Blum, A. L. and Rivest, R. L. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. Streaming variational Bayes. In *Advances in Neural Information Processing Systems*, pp. 1727–1735, 2013.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018.

Diethe, T., Borchert, T., Thereska, E., Pigem, B. d. B., and Lawrence, N. Continual learning in practice. In *NeurIPS 2018 Workshop on Continual Learning*, 2018.

Farquhar, S. and Gal, Y. A unifying Bayesian view of continual learning. In *NeurIPS 2018 workshop on Bayesian Deep Learning*, 2018.

Honkela, A. and Valpola, H. On-line variational Bayesian learning. In *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pp. 803–808, 2003.

Kamra, N., Gupta, U., and Liu, Y. Deep generative dual memory network for continual learning. *arXiv preprint arXiv:1710.10368*, 2017.

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.

Lauritzen, S. L. Time series analysis in 1880: A discussion of contributions made by T.N. Thiele. *International Statistical Review/Revue Internationale de Statistique*, pp. 319–331, 1981.

Lomonaco, V., Maltoni, D., and Pellegrini, L. Fine-grained continual learning. *arXiv preprint arXiv:1907.03799*, 2019.

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

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

McInerney, J., Ranganath, R., and Blei, D. The population posterior and Bayesian modeling on streams. In *Advances in neural information processing systems*, pp. 1153–1161, 2015.

Moreno-Munoz, P., Artés-Rodriguez, A., and A. Álvarez, M. Continual multi-task Gaussian processes. *arXiv preprint arXiv:1911.00002*, 2019.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. Variational continual learning. In *International Conference on Learning Representations*, 2017.

Opper, M. and Winther, O. A Bayesian approach to online learning. *On-line learning in neural networks*, pp. 363–378, 1998.

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

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Ritter, H., Botev, A., and Barber, D. Online structured Laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems*, pp. 3738–3748, 2018.

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., and Wayne, G. Experience replay for continual learning. In *Advances in Neural Information Processing Systems 32*, pp. 348–358. 2019.

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., Luketina, J., Czarnecki, W. M., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. Progress & compress: A scalable framework for continual learning. In *Proceedings of the Thirty-fifth International Conference on Machine Learning*, 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.

Theobald, T. On the frontiers of polynomial computations in tropical geometry. *Journal of Symbolic Computation*, 41(12):1360–1375, 2006.

Titsias, M. K., Schwarz, J., Matthews, A. G. d. G., Pascanu, R., and Teh, Y. W. Functional regularisation for continual learning using gaussian processes. *arXiv preprint arXiv:1901.11356*, 2019.

Tiwary, H. R. On the hardness of computing intersection, union and Minkowski sum of polytopes. *Discrete & Computational Geometry*, 40(3):469–479, 2008a.

Tiwary, H. R. *Complexity of some polyhedral enumeration problems*. PhD thesis, Saarländische Universität, 2008b.

Tseran, H., Khan, M. E., Harada, T., and Bui, T. D. Natural variational continual learning. In *NeurIPS 2018 Workshop on Continual Learning*, 2018.

van de Ven, G. M. and Tolias, A. S. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 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.