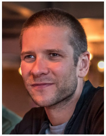
Optimal Continual Learning (CL) has Perfect Memory and is NP-hard

Jeremias Knoblauch[1,2], Hisham Husain[3,4], Tom Diethe[5]

November 6, 2020



[1]University of Warwick, Department of Statistics
[2]The Alan Turing Institute for Data Science and AI
[3]Australian National University, College of Engineering & Computer Science
[4]Data61, Sydney
[5]Amazon

Spotlight: Contributions

Theory:

(1) Connects set theory to *catastrophic forgetting* in CL

(2) Avoiding catastrophic forgetting ($=$ optimal CL)

- (A) is NP-HARD;
- (B) needs perfect memory.

Practical ramifications:

(A) CL algorithms $=$ heuristics for NP-HARD problem

(B) CL with memorization $>$ CL with regularization

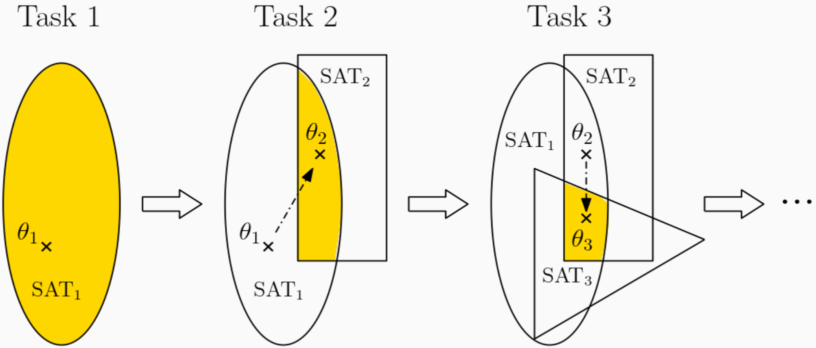
Spotlight: CL and set theory

No catastrophic forgetting $=$ all tasks $\textsc{Sat}$isfy optimality criterion $\mathcal{C}$

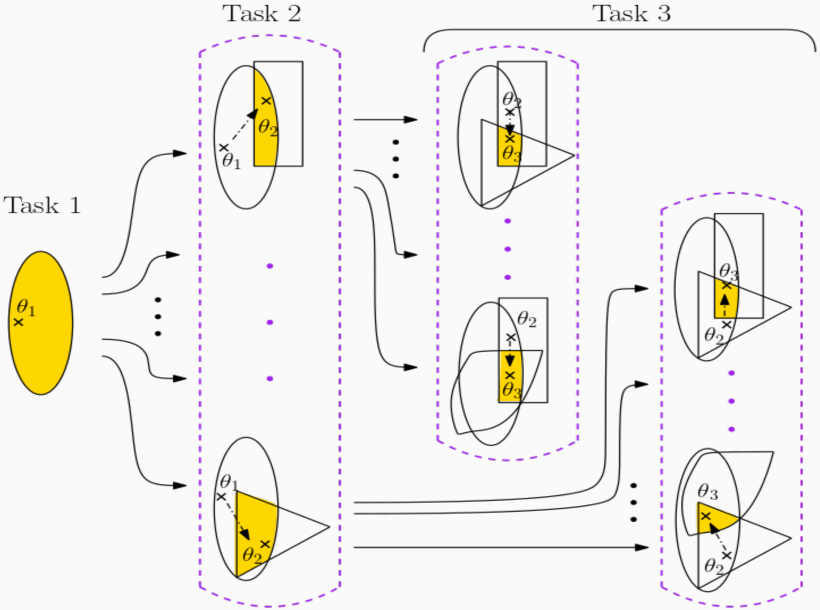
Interpretations using $\textsc{Sat}$isfiability sets:

(1) $\textsc{Sat}_t = \{\boldsymbol{\theta} \in \boldsymbol{\Theta} : \mathcal{C}(\boldsymbol{\theta})$ is satisfied on task $t\}$.

(2) **No** catastrophic forgetting ($=$ optimal CL) $\iff \boldsymbol{\theta}_t \in \cap_{i=1}^{t} \textsc{Sat}_i$.



Spotlight: Hardness and memory for CL



Structure

(1) Set Theory & CL

- (1.1) CL & Catastrophic Forgetting
- (1.2) Analyzing CL via sets

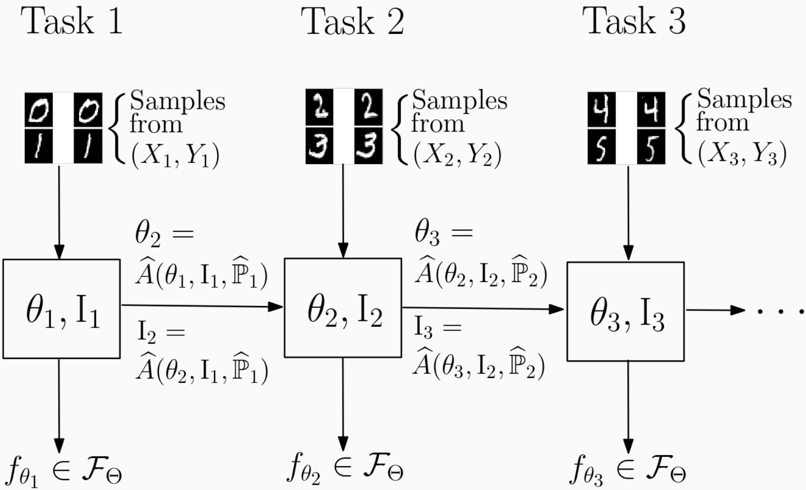
(2) Optimal CL: NP-hardness

- (2.1) The set intersection problem
- (2.2) NP-hardness results
- (2.3) A linear model example

(3) Optimal CL: Perfect Memory

- (3.1) Defining Perfect Memory
- (3.2) Memory requirements of CL
- (3.3) A linear model example

(4) Practical Ramifications: Memorization vs Regularization



Catastrophic Forgetting: θ_3 fails to correctly label Task 1, i.e. 

(1.2) Catastrophic Forgetting & Optimality

Notation:

$\boldsymbol{\theta}_t$ = parameter value after task t

$\mathcal{Q}$ = empirical distributions of all possible tasks

$\widehat{\mathbb{P}}_t \in \mathcal{Q}$ = t-th task's empirical distribution:

$$\widehat{\mathbb{P}}_t(x,y) = \frac{1}{n_t}\sum_{i=1}^{n_t} \delta_{(y_i^t, x_i^t)}(y,x), \text{ for } n_t \in \mathbb{N}, y_i^t \in \mathbb{R}, x_i^t \in \mathbb{R}^d$$

$\mathcal{C}$ = Optimality criterion. E.g., linear model & ε-error bound:

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

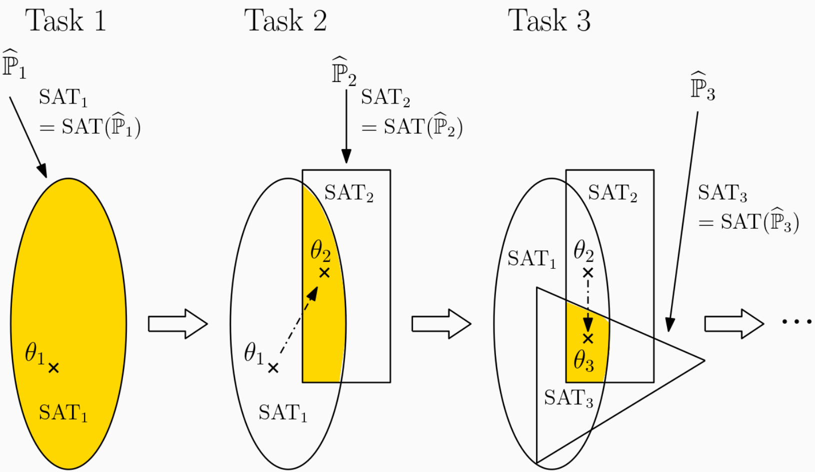
$\textsc{Sat}_t$ = $\textsc{Sat}$isfiability sets of task t, $\textsc{Sat}_t = \{\boldsymbol{\theta} \in \boldsymbol{\Theta} : \mathcal{C}(\boldsymbol{\theta}, \widehat{\mathbb{P}}_t) = 1\}$

(i) Optimality on all tasks = No catastrophic forgetting

(ii) Optimality $\iff \boldsymbol{\theta}_t \in \cap_{i=1}^{t} \textsc{Sat}_i$

(iii) Lemma 1: Analysis of optimal CL via $\textsc{Sat}_t$ valid!

Meaning of Lemma 1: Optimal CL has a set-theoretic interpretation



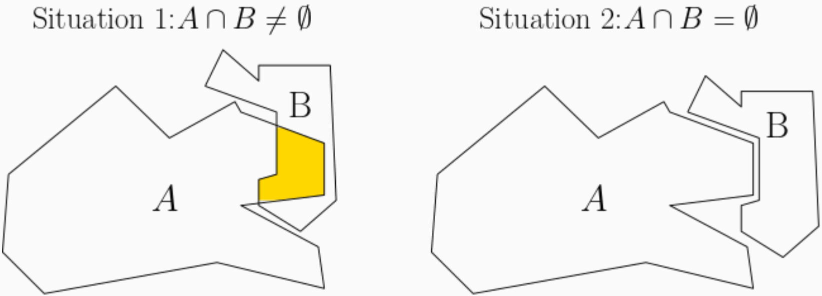
(2.1) The set intersection problem

Notation:

$$\textsc{Sat}_{\mathcal{Q}} = \text{set of all possible } \textsc{Sat}_t$$
$$\textsc{Sat}_{\cap} = \text{all finite intersections } \cap_{i=1}^{t} \textsc{Sat}_i \text{ of sets in } \textsc{Sat}_{\mathcal{Q}}$$

Lemma 2

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



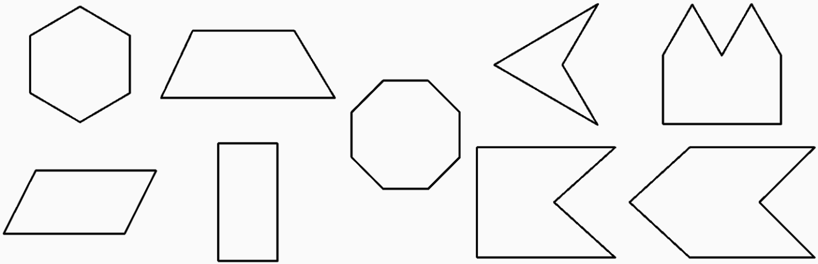
(2.2) NP-Hardness result

Theorem 1

If $\textsc{Sat}_{\mathcal{Q}} \supseteq S$ *or* $\textsc{Sat}_{\cap} \supseteq S$ *so that* S *is the set of tropical hypersurfaces or the set of polytopes on* $\mathbf{\Theta}$*, then optimal* CL *is* NP-$\textsc{hard}$*.*

Proof sketch.

Combine Lemma 2 with the fact that the intersection problem is NP-$\textsc{complete}$ if S is the set of tropical hypersurfaces or the set of polytopes on $\mathbf{\Theta}$. □



(2.3) A linear model example for NP-Hardness

(i) $\boldsymbol{\theta}$ = coefficient vector of linear model

(ii) $\mathcal{Q}$ = all tasks with empirical measures

$$\widehat{\mathbb{P}}_t(x, y) = \frac{1}{n_t} \sum_{i=1}^{n_t} \delta_{(y_i^t, x_i^t)}(y, x), \text{ for } n_t \in \mathbb{N}, y_i^t \in \mathbb{R}, x_i^t \in \mathbb{R}^d$$

(iii) $\mathcal{C}$ = ε-upper bound criterion in the L_1-norm:

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

$\implies$ $\textsc{Sat}_\cap$ contains *all* polytopes in $\boldsymbol{\Theta}$, since

$$\textsc{Sat}_t = \left\{ \boldsymbol{\theta} \in \boldsymbol{\Theta} : \boldsymbol{y}^t - \boldsymbol{\theta}^T \boldsymbol{X}^t \leq \varepsilon \cdot n_t \text{ and } \boldsymbol{\theta}^T \boldsymbol{X}^t - \boldsymbol{y}^t \geq \varepsilon \cdot n_t \right\}$$

(2.3) A linear model example for NP-Hardness

Q: So what? Who cares about simple Linear Models?

Corollary 1

If deciding whether $A \cap B = \emptyset$ for $A, B \in S$ is computationally at least as hard as for the collection of polytopes in $\mathbf{\Theta}$, optimal CL is NP-$\textsc{hard}$.

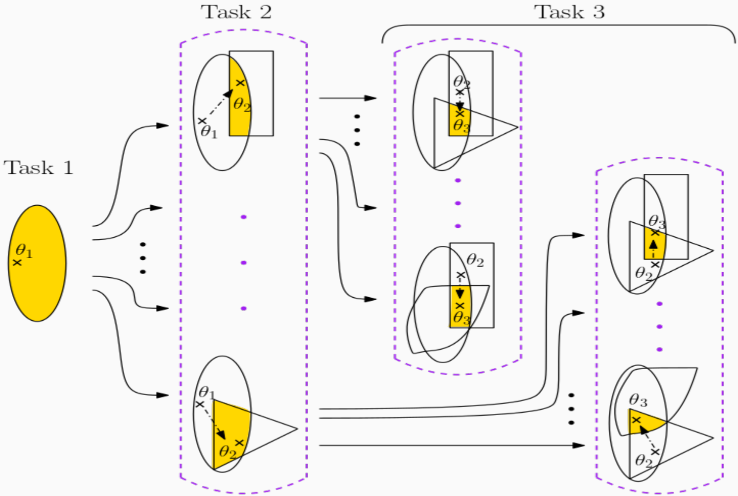
Meaning of Corollary 1:

Complicated nonlinear models / Neural Networks also NP-$\textsc{hard}$

(3.1) Defining Perfect Memory

Intuition: We could encounter *any* $\textsc{Sat}_t \in \textsc{Sat}_{\mathcal{Q}}$ at task t

$\implies$ Need all information in $\cap_{i=1}^{t-1}\textsc{Sat}_i$ from tasks $1, \ldots t-1$



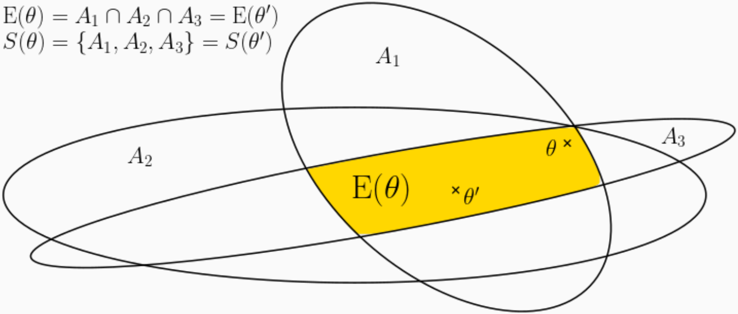
(3.1) Defining Perfect Memory

Q: Smallest necessary informational content in $\cap_{i=1}^{t}\text{SAT}_i$?
$\Longrightarrow$ The subset of $\cap_{i=1}^{t}\text{SAT}_i$ without *equivalent/redundant* elements!

Definition 1 (Equivalence set ($\approx$ 'redundance class'))

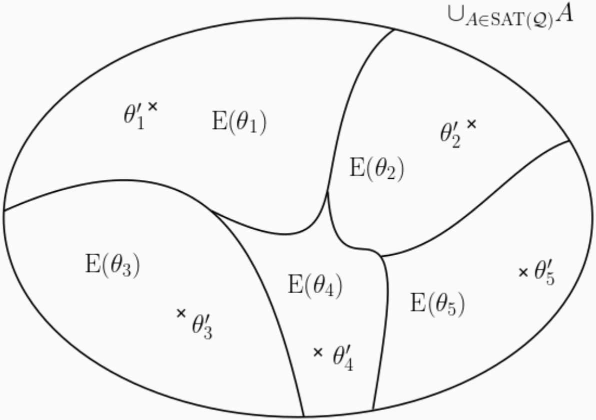
For $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, define $S(\boldsymbol{\theta}) = \{A \in \text{SAT}_{\mathcal{Q}} : \boldsymbol{\theta} \in A\}$ and the equivalence sets

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



(3.1) Defining Perfect Memory

Perfect Memory $=$ store element of each $\mathrm{E}(\boldsymbol{\theta})$ s.t. $\mathrm{E}(\boldsymbol{\theta}) \subseteq \cap_{i=1}^{t} \mathrm{SAT}_i$



$\mathrm{C} =$ Minimal Cover
$F =$ Minimal Representation

$I = \{1, 2, \ldots 5\}$
$f(i) = \theta_i', i \in I$
$\mathrm{C} = \{\mathrm{E}(\theta_i)\}_{i \in I}$
$F = \{f(i)\}_{i \in I}$

(3.2) CL Memory Requirements

Lemma 4

For optimal CL *algorithms, there exists* $h : \boldsymbol{\Theta} \times i \to 2^{\boldsymbol{\Theta}}$ *for which* $h(\boldsymbol{\theta}_t, \mathrm{I}_t) = C_t$ *is s.t.* $C_t \cap A = \emptyset \iff \cap_{i=1}^{t} \mathrm{S{\scriptsize AT}}_i \cap A = \emptyset$, $\forall A \in \mathrm{S{\scriptsize AT}}_{\mathcal{Q}}$.

Meaning of Lemma 4:

Optimal CL memorizes enough to solve set intersection problem

Corollary 2

If $\mathcal{C}$ *and* $\mathcal{Q}$ *are s.t.* $\mathrm{E}(\boldsymbol{\theta}) \in \mathrm{S{\scriptsize AT}}_{\mathcal{Q}}$ *for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, *any optimal* CL *algorithm has perfect memory.*

Meaning of Corollay 2:

You can only solve the set intersection problem with perfect memory

(3.3) A linear model example for Perfect Memory

(i) $\boldsymbol{\theta}$ = coefficient vector of linear model

(ii) $\mathcal{Q}$ = all tasks with empirical measures

$$\widehat{\mathbb{P}}_t(x, y) = \frac{1}{n_t} \sum_{i=1}^{n_t} \delta_{(y_i^t, x_i^t)}(y, x), \text{ for } n_t \in \mathbb{N}, y_i^t \in \mathbb{R}, x_i^t \in \mathbb{R}^d$$

(iii) $\mathcal{C}$ = ε-upper bound criterion in the L_1-norm:

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

$\Longrightarrow$ $\mathrm{S}\small{\mathrm{AT}}_{\mathcal{Q}}$ contains $\{\boldsymbol{\theta}\} = \mathrm{E}(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.
(Take $\boldsymbol{X}^t, \boldsymbol{y}^t$ s.t. $\boldsymbol{y}^t - \varepsilon n_t = \boldsymbol{\theta}^T \boldsymbol{X}^t$ is solved by **unique** $\boldsymbol{\theta} \in \boldsymbol{\Theta}$)

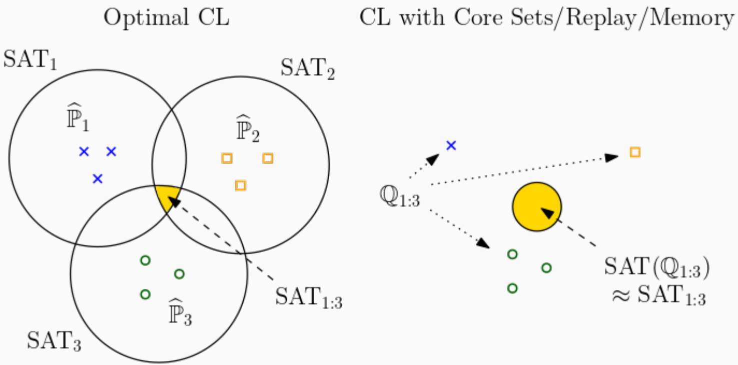
(4) Practical Ramifications: Memorization vs Regularization

Implications for algorithm design:

(1) Optimal CL = moving $\cap_{i=1}^{t-1}\textsc{Sat}_i \longrightarrow \cap_{i=1}^{t-1}\textsc{Sat}_i \cap \textsc{Sat}_t$

(2) We need perfect memory to do that without errors

$\Rightarrow$ memorization should work better than regularization



Summary

Theory:

(1) Connects set theory to *catastrophic forgetting* in CL

(2) Avoiding catastrophic forgetting ($=$ optimal CL)

  (A) is NP-HARD;

  (B) needs perfect memory.

Practical ramifications:

(A) CL algorithms $=$ heuristics for NP-HARD problem

(B) CL with memorization $>$ CL with regularization

Contact me/follow my research at

E-mail address: j.knoblauch@warwick.ac.uk

Personal Website: https://jeremiasknoblauch.github.io/

Twitter Handle: @LauchLab