Three types of incremental learning

Gido M. van de Ven [1,2,3] Tinne Tuytelaars [3] Andreas S. Tolias [1,4]

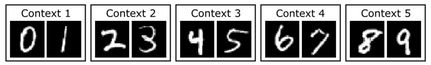
[1]Baylor College of Medicine [2]University of Cambridge [3]KU Leuven [4]Rice University

Overview

Based on how the non-stationary aspect of the data relates to the mapping to learn, we identify three fundamental types — or 'scenarios' — of continual learning:

- **Task-incremental learning (Task-IL):** incrementally learn a set of clearly distinct tasks;
- **Domain-incremental learning (Domain-IL):** learn to solve same kind of problem in different contexts;
- **Class-incremental learning (Class-IL):** incrementally learn to distinguish a growing number of classes.

Split MNIST example



	Question at test time
Task-incremental learning	Choice between two digits of same context
Domain-incremental learning	Is the digit odd or even?
Class-incremental learning	Choice between all ten digits

Formalization in the 'academic continual learning setting'

Academic continual learning setting: a classification problem split up into non-overlapping parts (which we call 'contexts') that are encountered sequentially.

Express each sample as consisting of three components: an input $\boldsymbol{x} \in \mathcal{X}$, a within-context label $y \in \mathcal{Y}$ and a context label $c \in \mathcal{C}$. The three scenarios are then defined based on how the context space $\mathcal{C}$ relates to the mapping to learn (see table on the right). Note that for class-incremental learning the mapping to learn can also be written as $f\colon \mathcal{X} \rightarrow \mathcal{G}$, with $\mathcal{G}$ the 'global label space' obtained by combining $\mathcal{C}$ and $\mathcal{Y}$.

Task-incremental learning	$f\colon \mathcal{X} \times \mathcal{C} \rightarrow \mathcal{Y}$
Domain-incremental learning	$f\colon \mathcal{X} \rightarrow \mathcal{Y}$
Class-incremental learning	$f\colon \mathcal{X} \rightarrow \mathcal{C} \times \mathcal{Y}$

Another way to distinguish these three scenarios is by whether, at test time, context identity is known and, if it is not, whether it must be inferred.

Generalization to more flexible, 'task-free' settings

Introduce a distinction between:

- **Context set:** collection of underlying distributions, denoted by $\{\mathcal{D}_c\}_{c \in \mathcal{C}}$;
- **Data stream:** sequence of experiences $e_1, e_2, \ldots$ presented to the algorithm.

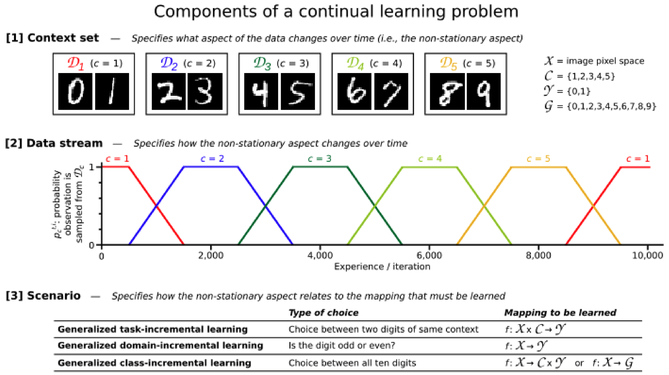
Every observation in each experience can be sampled from any combination of underlying datasets from the context set::

$$e_t[i] \sim \sum_{c \in \mathcal{C}} p_c^{t,i} \mathcal{D}_c \qquad (1)$$

whereby $e_t[i]$ is observation i of experience t and $p_c^{t,i}$ is the probability that this observation is sampled from $\mathcal{D}_c$.

Importantly, from a probabilistic perspective, this means two observations at different points in time can only differ w.r.t. the context(s) they are sampled from. The context space $\mathcal{C}$ thus describes the non-stationary aspect of the data.

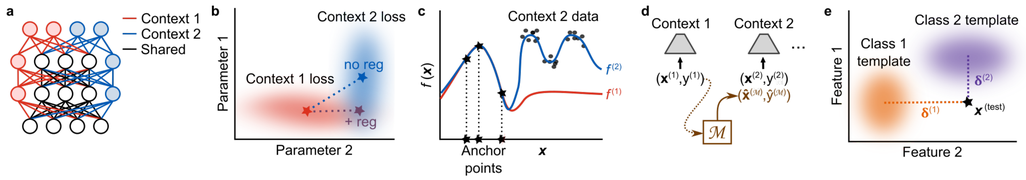
Generalized versions of the three scenarios can be defined as before, based on how the context space $\mathcal{C}$ relates to the mapping to learn.



Discussion

- Continual learning is not a unitary problem: we describe three fundamentally different scenarios, each with their own challenges;
- The three continual learning scenarios differ substantially in terms of difficulty and in terms of the effectiveness of different strategies;
- In the real-world, continual learning problems are often complex and 'mixtures' of these scenarios (see the paper): we believe it can be useful to approach such problems as consisting of a combination of these three fundamental types of incremental learning.

Strategies for continual learning



a **Context-specific components** uses certain parts of the network only for specific contexts;

b **Parameter regularization** encourages parameters important for past contexts not to change too much when learning new contexts;

c **Functional regularization** encourages the input-output mapping learned previously not to change too much at a particular set of inputs (the 'anchor points') when training on new contexts;

d **Replay** complements the training data of a new context with data representative of past contexts;

e **Template-based classification** learns a template for each class (e.g., a prototype or generative model) and classifies based on which template is most suitable for the sample to be classified.

Empirical comparison

Strategy	Method	Budget	GM	Task-IL	Domain-IL	Class-IL
Baselines	*None – lower target*			*84.32 (± 0.99)*	*60.13 (± 1.66)*	*19.89 (± 0.02)*
	Joint – upper target			*99.67 (± 0.03)*	*98.59 (± 0.05)*	*98.17 (± 0.04)*
Context-specific components	Separate Networks	-	-	99.57 (± 0.03)	-	-
	XdG	-	-	99.10 (± 0.10)	-	-
Parameter regularization	EWC	-	-	99.06 (± 0.15)	63.03 (± 1.58)	20.64 (± 0.52)
	SI	-	-	99.20 (± 0.11)	66.94 (± 1.13)	21.20 (± 0.57)
Functional regularization	LwF	-	-	99.60 (± 0.03)	71.18 (± 1.42)	21.89 (± 0.32)
	FROMP	100	-	99.12 (± 0.13)	84.86 (± 1.02)	77.38 (± 0.64)
Replay	DGR	-	yes	99.50 (± 0.03)	95.57 (± 0.30)	90.35 (± 0.24)
	BI-R	-	yes	99.61 (± 0.03)	97.26 (± 0.15)	94.41 (± 0.15)
	ER	100	-	98.98 (± 0.07)	93.75 (± 0.24)	88.79 (± 0.20)
	A-GEM	100	-	98.54 (± 0.10)	87.67 (± 1.33)	65.10 (± 3.64)
Template-based classification	Generative Classifier	-	yes	-	-	93.82 (± 0.06)
	iCaRL	100	-	-	-	92.49 (± 0.12)

Results on Split MNIST in the academic continual learning setting. Final test accuracy (as %, averaged over all contexts) is reported. 'Budget' indicates the number of examples per class stored in a memory buffer, 'GM' indicates whether a generative model was learned using additional network capacity. Each experiment was run 20 times, reported is the mean (± SEM). See the paper for more experimental comparisons (e.g., Split CIFAR-100 and a 'task-free' version of Split MNIST).