Learning from Data Streams versus (Online) Continual Learning

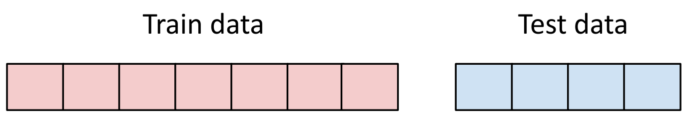
Bernhard Pfahringer

Te Ipu o te Mahara
Artificial Intelligence Institute
THE UNIVERSITY OF WAIKATO

Outline

- **Stream Learning (SL)**
- **Continual Learning (CL)**
- **Online Continual Learning (OCL)**
- **Synthesis?**

Batch Learning vs Stream Learning (SL)

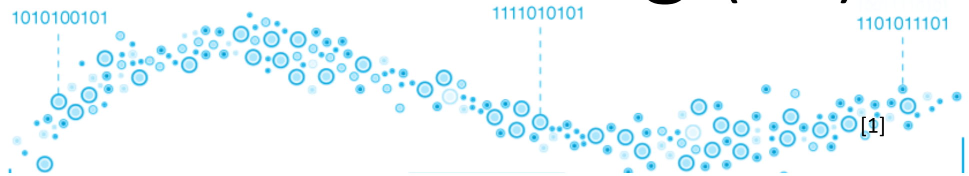


Assumes **data is IID**

Uses a **large** amount of **computing** resources to train the model.

Can only **predict** after (**extensive**) **training**.

If the *underlying data distribution changes* (**concept drift**)
➔ **re-train** the model.



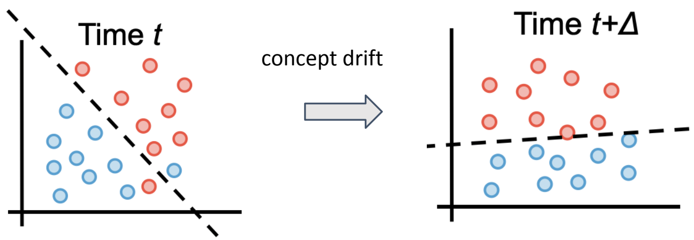
Assumes **data is non-IID**

Incrementally online learn form instance/mini-batch at a time.

Should use **limited computing** resources.

Able to **predict** at **any given moment**.

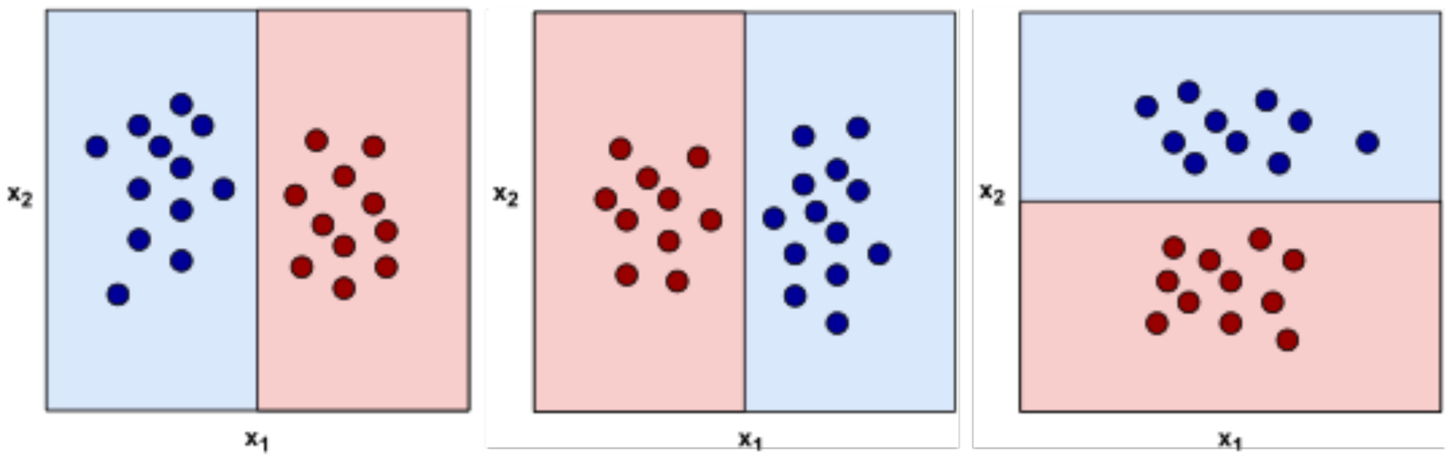
Must **adapt** to **concept drifts online**.



[1] source: https://www.onaudience.com/resources/what-is-data-stream-and-how-to-use-it/ [2] Gomes, H., Montiel, J., & Bifet, A. (2020). Data Stream Mining COMP523-2020(HAM)

Concept Drift (types)

- ***Effect on the decision boundary*** (impact):
    - real and virtual concept drifts.

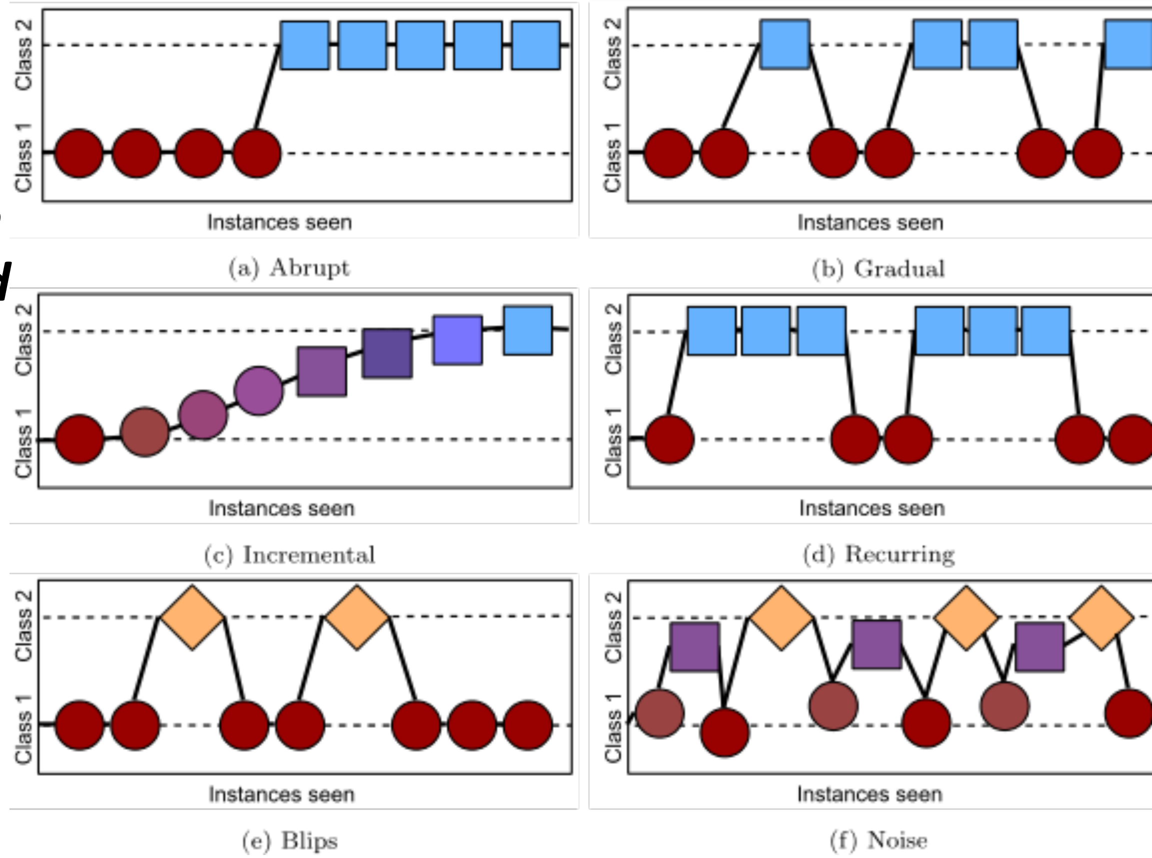


(a) Original data (b) Virtual Drift (c) Real Drift

(Sua´rez-Cetrulo et al., 2023)

Concept Drift (types)

- ***Evolution of the relationship between features and the target and the speed of change***:
  - abrupt (sudden), gradual, and incremental drifts
- ***Recurrent concept drifts***:
  - particular data distribution reoccurs in the stream
- **Random blips/outliers/noise**:
  - few instances which do not belong to the current distribution popup in the stream for a very short period of time



(Suarez-Cetrulo et al., 2023)

Drift Detectors

- ***Methods based on differences between two distributions***:
  - ADaptive sliding WINdow (ADWIN) [Bifet and Gavalda, 2007]
- ***Methods based on sequential analysis***:
  - methods founded on the Sequential Probability Ratio Test (SPRT)[Wald, 1947].
  - CUSUM and Page–Hinkley Test [Page, 1954]
- ***Methods based on statistical process control***:
  - consider the classification problem as a statistical process to monitor the evolution of some performance indicators like error rate to apply heuristics to find change points.
  - DDM [Gama et al., 2004]
  - EDDM [Baena-Garcıa et al., 2006]

SL Methods

- ***Classification***:
  - Naive Bayes (**NB**), Hoeffding Tree (**HT**) [Hulten et al., 2001] Adaptive Random Forest (**ARF**) [Gomes et al., 2017a], Streaming Random Patches (**SRP**) [Gomes et al., 2019], **CAND**[Gunasekara et al., 2022c]
- ***Regression***:
  - **FIMT-DD** [Ikonomovska *et al.*, 2011], Adaptive Random Forest Regressor (**ARF-REG**) [Gomes et al., 2018], **SOKNL** [Sun et al., 2022]
- ***Clustering***:
  - **CluStream** [Aggarwal et al., 2003], **Adaptive Streaming k-Means** [Puschmann et al, 2017]

[Aggarwal et al., 2003] Aggarwal CC, Han J, Wang J, Yu PS (2003) A framework for clustering evolving data streams. In: Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29, VLDB '03, pp 81–92

[Puschmann et al, 2017] Puschmann D, Barnaghi P, Tafazolli R (2017) Adaptive clustering for dynamic iot data streams. IEEE Internet of Things Journal 4(1):64–74

Evaluation

- *Methods*
  - ***test-then-train (prequential evaluation)*** [Gama et al., 2013].
  - ***prequential evaluation*** *with* ***a sliding window, or a fading factor*** [Gama et al., 2013]
    - to gracefully forget the performance on instances from the distant past
  - ***Data stream cross-validation*** [Bifet et al., 2015]
    - models are trained and tested in parallel on different folds of the data.
  - ***Continuous re-evaluation*** [Grzenda et al., 2020a; Grzenda et al., 2020b]
    - considers the **verification latency** in the streaming setting with **partially *delayed*** labels.
    - evaluates how **fast** a model can **transform** from an **initial possibly incorrect** prediction to a **correct** prediction **prior true label availability**.
- Metrics (other than accuracy)
  - ***sensitivity and specificity***
    - for imbalanced data streams [Bahri et al., 2021].
  - ***Kappa statistic***
    - compares the model's prequential accuracy **against the chance classifier** [Bifet et al., 2018].
  - ***Kappa M***
    - compares the current model's performance **against the majority class classifier** [Bifet et al., 2018].
  - ***Kappa temporal***
    - compares the current model's performance **against a ”no- change” model** [Bifet et al., 2018].

One big issue: Labelling of data streams

I. **Immediate** and **fully** labelled,
II. **Delayed** and **fully** labelled,
III. **Immediate** and **partially** labelled,
IV. **Delayed** and **partially** labelled.

- (i) default assumption, but naïve
- (ii) common in automatic (numeric) prediction, e.g. river levels, …
- (iii) semi-supervised SL, use cases???
- (iv) common in business processes, e.g. mortgage approval

Life Long Learning

- Thrun & Mitchell 1995: Lifelong robot learning

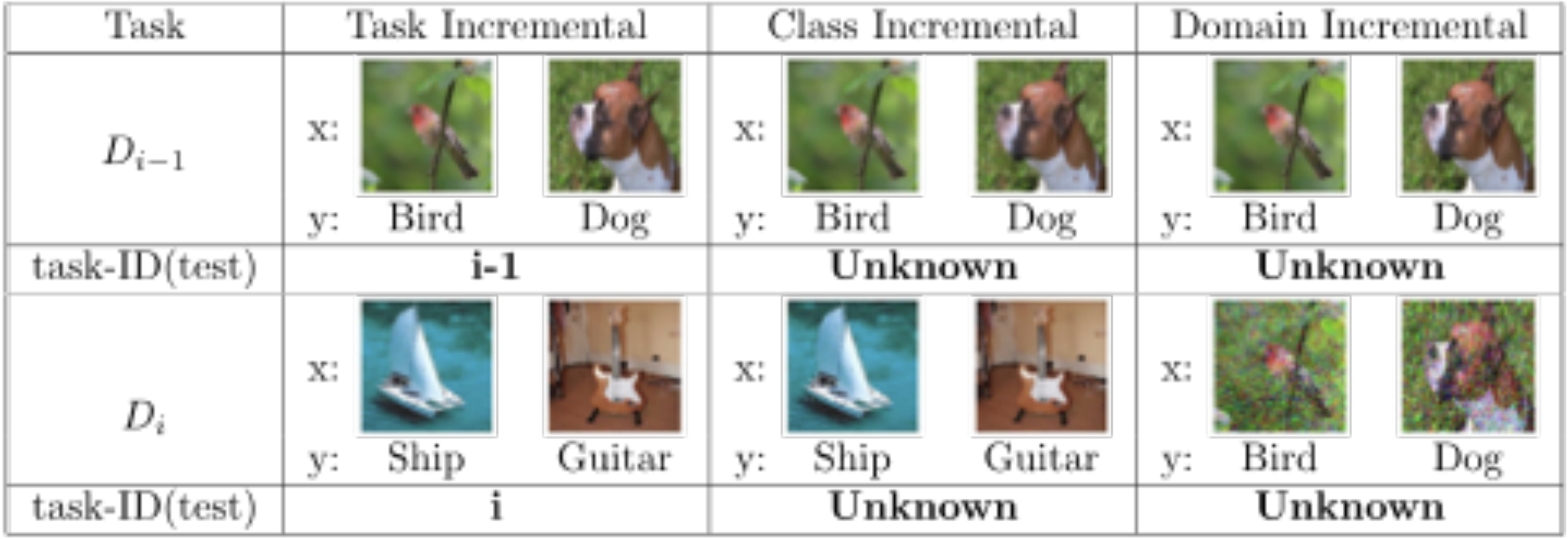
- More than one task
- Generalize across tasks
- Dependent and independent tasks
- Transfer learning

Continual Learning (CL)

”… to learn a model for a
large number of tasks sequentially
without forgetting knowledge obtained from the preceding tasks,
where the data in the old tasks are not available
anymore during training new ones”

from https://paperswithcode.com/task/continual-learning
[11-Sept-2023: 631 papers with code • 24 benchmarks • 28 datasets]

CL settings



Task	Task Incremental	Class Incremental	Domain Incremental
D_{i-1}	x: y: Bird Dog	x: y: Bird Dog	x: y: Bird Dog
task-ID(test)	**i-1**	**Unknown**	**Unknown**
D_i	x: y: Ship Guitar	x: y: Ship Guitar	x: y: Bird Dog
task-ID(test)	**i**	**Unknown**	**Unknown**

(Mai et al., 2022)

Evaluation

On a stream with ***T*** tasks, after training in tasks **1 to *i***, let $\mathbf{a}_{i,j}$ be the accuracy on the held-out test set for **task *j***.

- ***Average accuracy*** (A_i) **at task *i***: represents the average accuracy by the **end of training** task ***i*** with the whole data sequence **up to *i***.

a	***te₁***	***te₂***	**…**	***te*$_{T-1}$**	***te*$_T$**
$\boldsymbol{tr_1}$	$a_{1, 1}$	$a_{1, 2}$	…	$a_{1, T-1}$	$a_{1, T}$
$\boldsymbol{tr_2}$	$a_{2, 1}$	$a_{2, 2}$	…	$a_{2, T-1}$	$a_{2, T}$
…	…	…	…	…	…
$\boldsymbol{tr_{T-1}}$	$a_{T-1, 1}$	$a_{T-1, 2}$	…	$a_{T-1, T-1}$	$a_{T-1, T}$
$\boldsymbol{tr_T}$	$a_{T, 1}$	$a_{T, 2}$	…	$a_{T, T-1}$	$a_{T, T}$

(Mai et al., 2022)

Evaluation

- ***Average forgetting*** (F_i) **at task *i*** : represents how much the model has **forgotton about task j** after being **trained** on **task** *i*. Compared against the **maximum accuracy** up to *i.*

- ***Backward Transfer*** (BWT): The positive influence of learning a new task on **previous** tasks' performance.

- **Forward Transfer** (FWT): The positive influence of learning a given task on **future** tasks' performance .

(Mai et al., 2022)

Backward transfer

a	***te***$_{1}$	***te***$_{2}$	**...**	***te***$_{T\text{-}1}$	***te***$_{T}$
tr$_{1}$	$a_{1,\,1}$	$a_{1,\,2}$	...	$a_{1,\,T-1}$	$a_{1,\,T}$
tr$_{2}$	$a_{2,\,1}$	$a_{2,\,2}$	...	$a_{2,\,T-1}$	$a_{2,\,T}$
...	...	...	...	...	...
tr$_{T\text{-}1}$	$a_{T-1,\,1}$	$a_{T-1,\,2}$	...	$a_{T-1,\,T-1}$	$a_{T-1,\,T}$
tr$_{T}$	$a_{T,\,1}$	$a_{T,\,2}$	...	$a_{T,\,T-1}$	$a_{T,\,T}$

(Mai et al., 2022)

Forward transfer

a	te_1	te_2	…	te_{T-1}	te_T
tr_1	$a_{1, 1}$	$a_{1, 2}$	…	$a_{1, T-1}$	$a_{1, T}$
tr_2	$a_{2, 1}$	$a_{2, 2}$	…	$a_{2, T-1}$	$a_{2, T}$
…	…	…	…	…	…
tr_{T-1}	$a_{T-1, 1}$	$a_{T-1, 2}$	…	$a_{T-1, T-1}$	$a_{T-1, T}$
tr_T	$a_{T, 1}$	$a_{T, 2}$	…	$a_{T, T-1}$	$a_{T, T}$

(Mai et al., 2022)

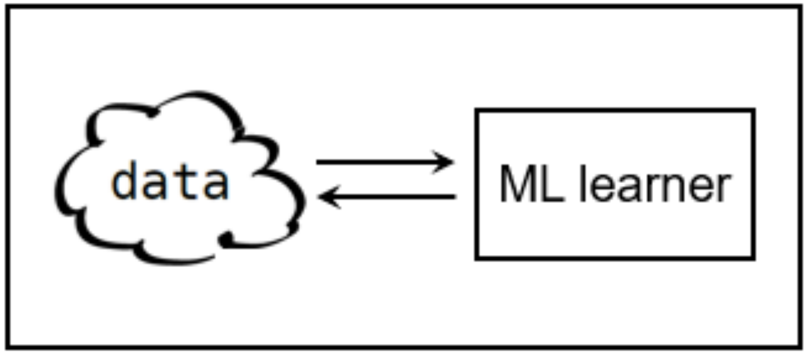
Methods

- ***Regularization***: **adjust the weights of the network** to **minimize the overwriting** of the weights for the **old** concept.
  - EWC [Kirkpatrick et al., 2017]
  - LWF [Li and Hoiem, 2017]

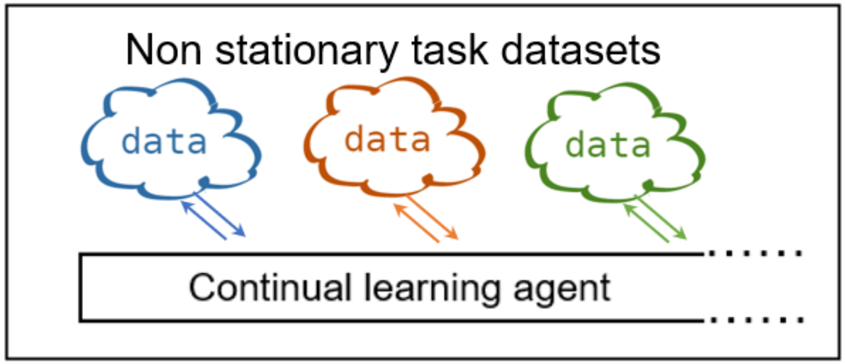
- ***Replay***: present a **mix of old and current concept's instances** to the NN based on a given policy while training.
  - GDUMB [Prabhu et al., 2020], ER [Chaudhry et al., 2019], MIR [Aljundi et al.,2019], REMIND [Hayes et al., 2020]
  - **Privacy concerns due to replay buffer** in some settings [Armstrong and Clifton, 2021; Mai et al., 2022]

- ***Parameter-isolation***: **avoid interference** by allocating **separate parameters** for **each task**.
  - *Fixed architecture*: only activates the relevant part of the network without changing the NN architecture
  - Dynamic architecture: adds new parameters for the new task while keeping the old parameters

Online continual learning

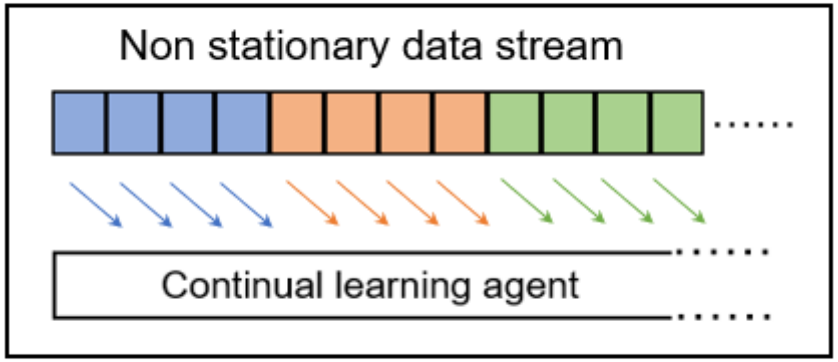


Standard machine learning



(Offline) Continual learning:
sequence of batch learning tasks

Online continual learning:
✔ Maintain past knowledge
✔ Accumulate new knowledge
✔ Single pass through data

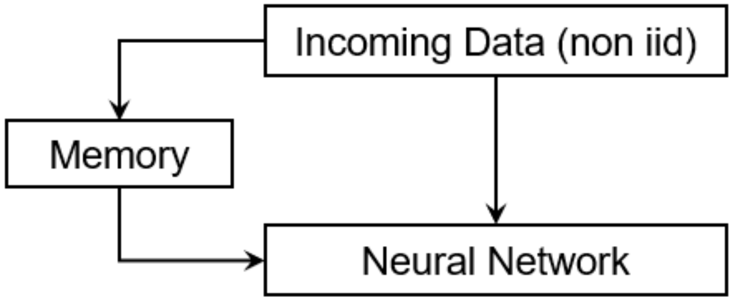


(Online) Continual learning

Rehearsal-based continual learning

- Rehearsal-based continual learning
  - Different variants of ER : ER, MIR, ASER, SCR, DER etc
  - Achieves state-of-the-art performance in a number of standard OCL benchmarks
  - Faces the challenge of memory overfitting

Experience Replay (ER)



$$\theta \leftarrow \mathrm{SGD}(B \cup \tilde{B}, \theta)$$

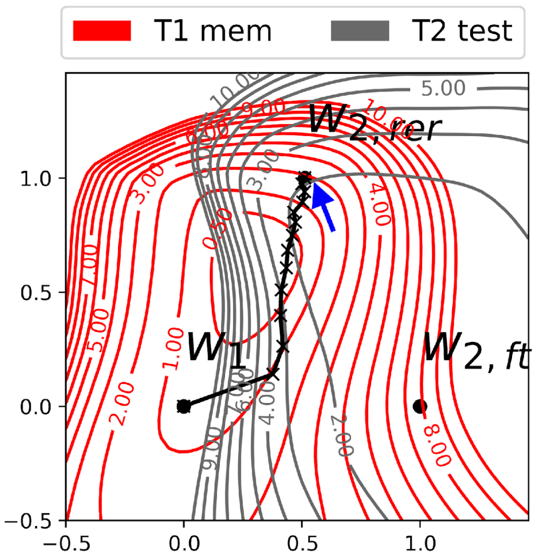
- Research question:
  - how to effectively perform rehearsal with the memorized samples in online continual learning

Loss Landscape Analysis: underfitting-overfitting dilemma

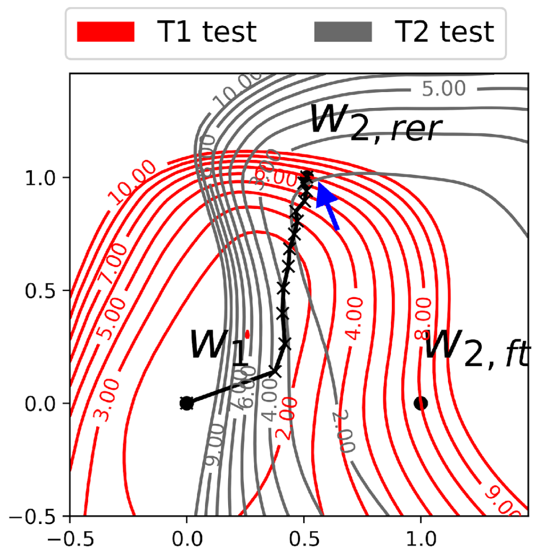
- Is Repeated Rehearsal (with k iterations) a good idea?

$$\theta_{t,k+1} = \theta_{t,k} - \frac{\eta}{|\mathcal{B}_t|} \sum_{\mathbf{x},y \in \mathcal{B}_t} \nabla \mathcal{L}\left(f_{\theta_{t,k}}(\mathbf{x}), y\right) - \frac{\eta}{\left|\mathcal{B}^{\mathcal{M}}_{t,k}\right|} \sum_{\mathbf{x},y \in \mathcal{B}^{\mathcal{M}}_{t,k}} \nabla \mathcal{L}\left(f_{\theta_{t,k}}(\mathbf{x}), y\right)$$

- The dilemma of overfitting locally and underfitting globally in online continual rehearsal



Loss on memory data: 2.1



Loss on test data: 7.9

Empirical Risk Minimization in Online Rehearsal

- What we want the CL method to do:

$$\min_{\theta} \mathcal{R}(\theta) = \frac{1}{\sum_t |\mathcal{B}_t|} \sum_t \sum_{\mathbf{x}, y \in \mathcal{B}_t} \mathcal{L}\left(f_\theta(\mathbf{x}), y\right)$$

- What the rehearsal-based CL method actually does: ERM for online rehearsal

$$\mathcal{R}_t(\theta) = \sum_{\mathbf{x}, y \in \mathcal{D}_{\mathcal{T}}} \mathcal{L}(f_\theta(\mathbf{x}), y) + \beta_t \lambda \sum_{\mathbf{x}, y \in \mathcal{D}^0_{\mathcal{M}}} \mathcal{L}(f_\theta(\mathbf{x}), y)$$

where $\lambda := \frac{|\mathcal{D}_{\mathcal{T}}|}{|\mathcal{D}^0_{\mathcal{M}}|}$ and $\beta_t := 1/(1 + \frac{2N^t_{cur}}{N^{\mathcal{T}}_{past}})$

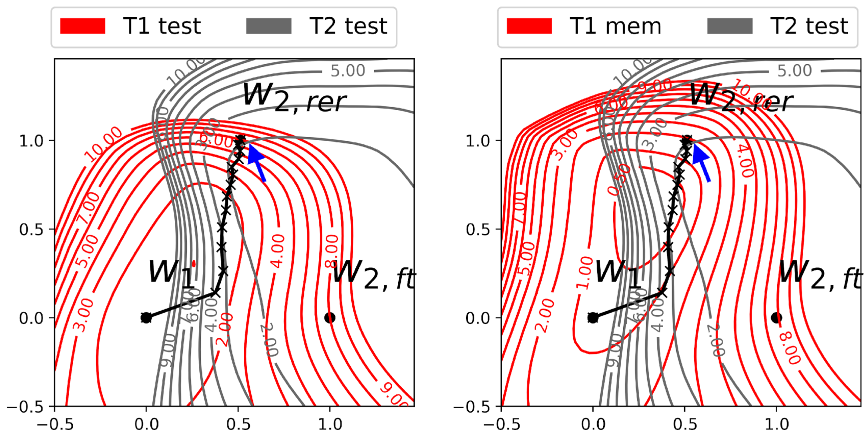
  - Bias
  - Problem-dependence
  - Dynamic

Proposed method: Repeated Augmented Rehearsal (RAR)

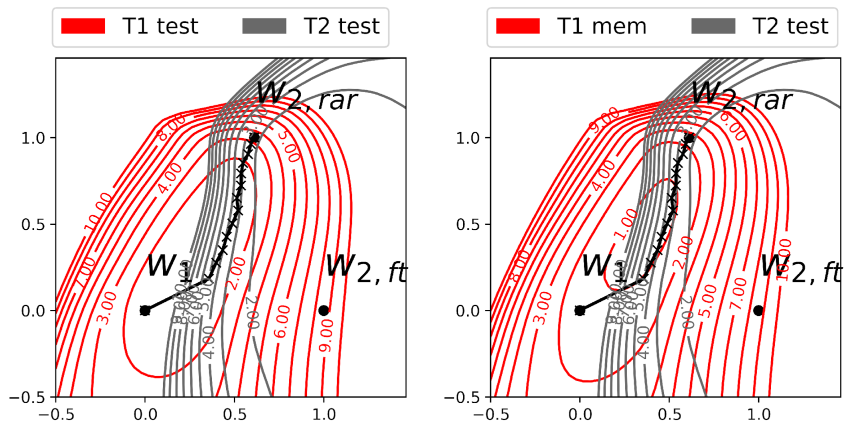
- Augmented Empirical Risk

$$\bar{\mathcal{R}}_t(\theta) = \sum_{\mathbf{x},y\in\mathcal{D}_{\mathcal{T}}} \int_G \mathcal{L}(f_\theta(g\mathbf{x}), y)d\mathbb{Q}(g) + \beta_t\lambda \sum_{\mathbf{x},y\in\mathcal{D}_{\mathcal{M}}} \int_G \mathcal{L}(f_\theta(g\mathbf{x}), y)d\mathbb{Q}(g)$$

Similar to i.i.d. learning setting, can reduce both the variance and generalization error.



Repeated Rehearsal

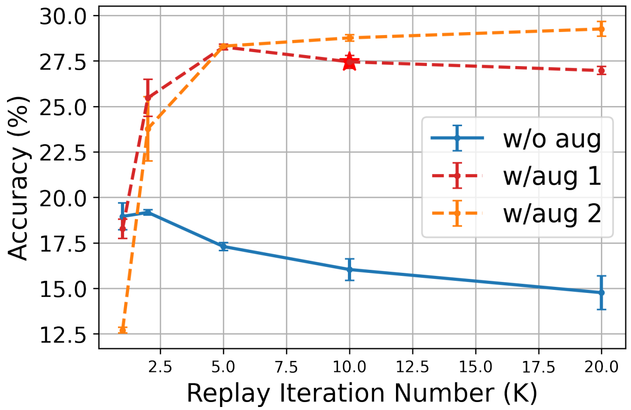


Repeated Augmented Rehearsal (RAR)

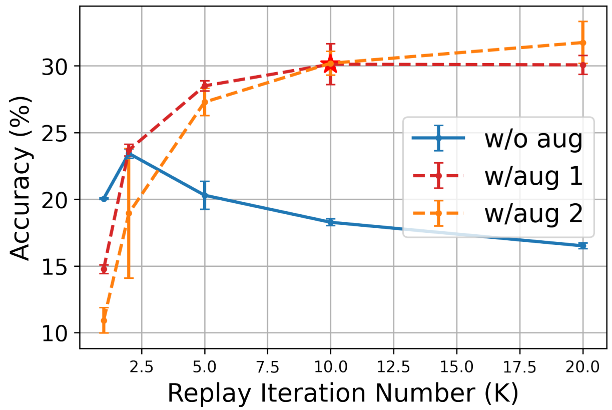
Experiments: Ablation Studies

- Interplay between Repeated and Augmented Rehearsal
  - Augmentation alone does not work well
  - Repeated rehearsal alone does not work well

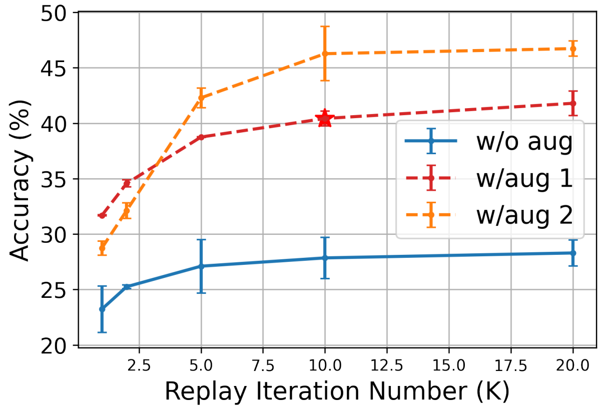
- RAR's Robustness to Large Numbers of Repeats



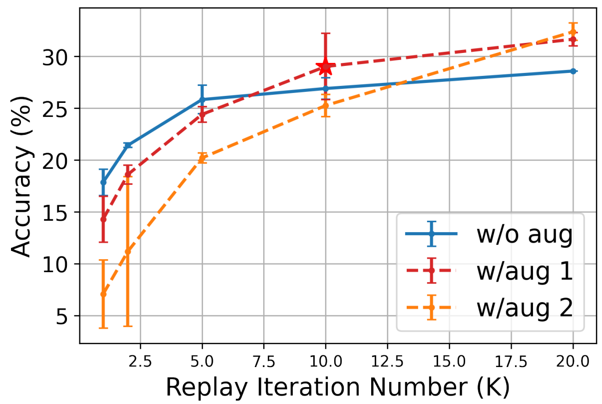
Seq-CIFAR100



Seq-MiniImageNet



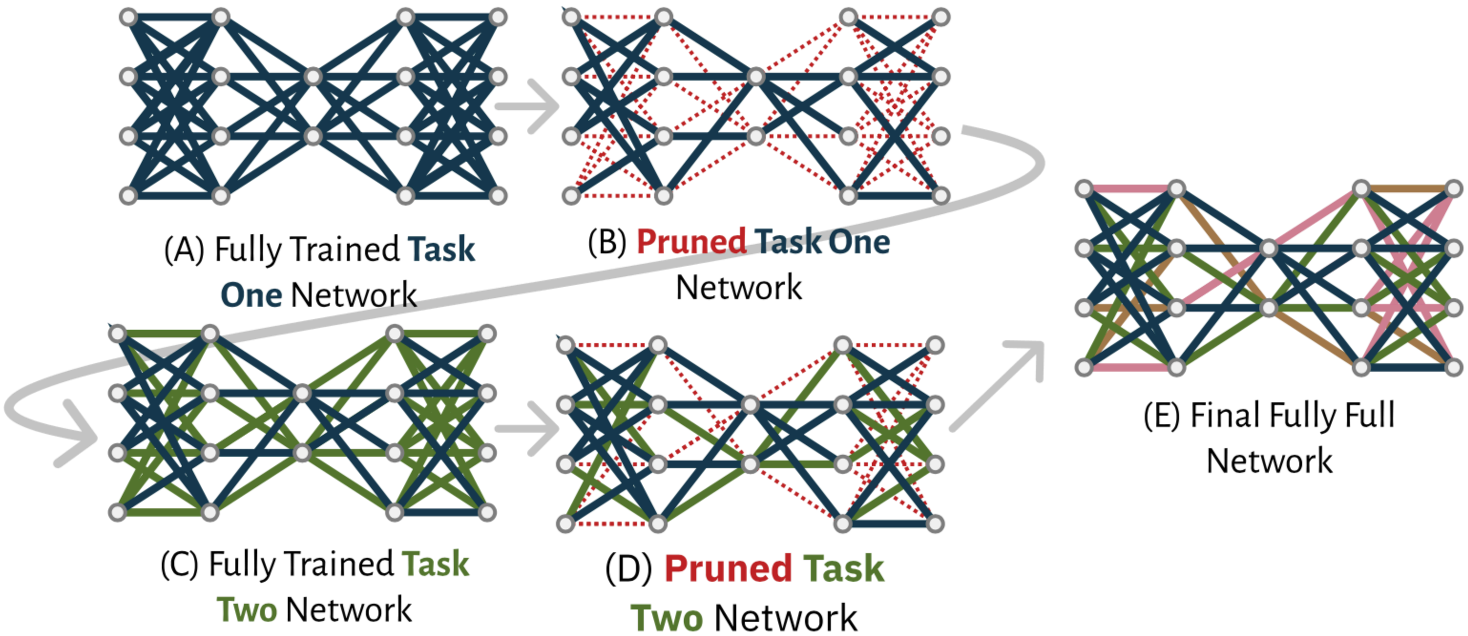
CORE50



CLRS

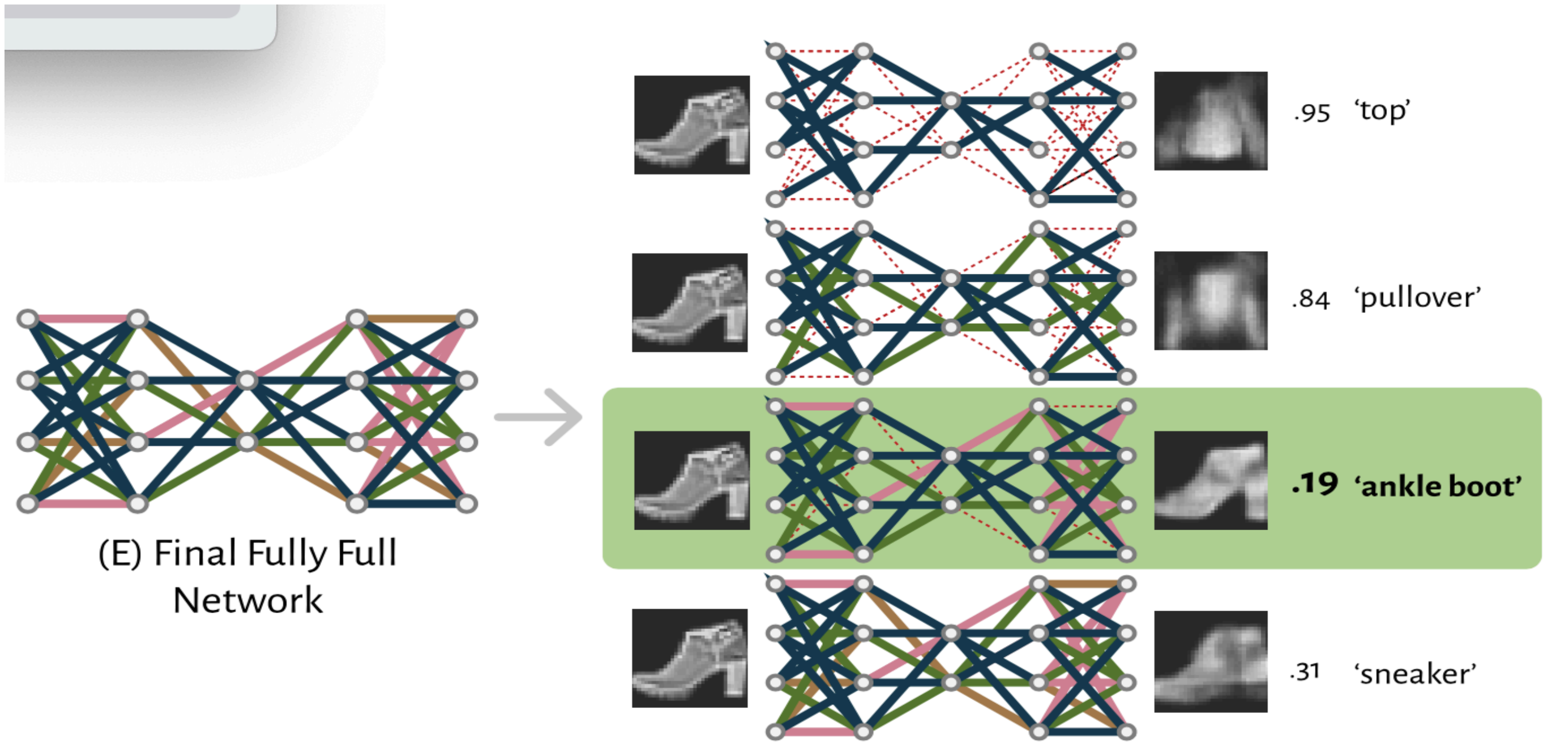
SurpriseNet: Anomaly Detection Inspired Class Incremental Learning

- Pruning-based with an additional AutoEncoder for task detection



SurpriseNet:

- For prediction, use reconstruction error to decide on task

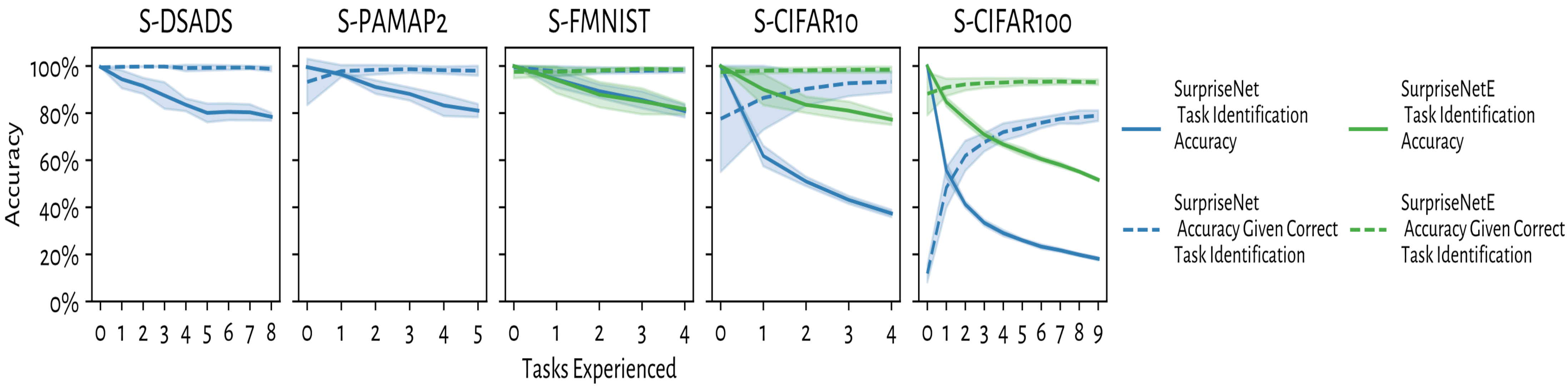


(E) Final Fully Full Network

(F) Task Inference

SurpriseNetE:

- Uses a pre-trained feature extractor for images



SL vs. OCL

- Stream Learning: **quickly** adjust to the **current concept** only

- Online Continual Learning has two learning objectives:
  - adjust to the **current concept**
  - **preserve knowledge** of previous concepts

- Both assume data is **non-IID**

- OCL: some methods need explicit **Task ID** (end of concept signal) for **training**

More on SL versus OCL:

Topic	SL	OCL
Setting	Single learning objective: adjust to current concept efficiently.	Dual learning objective: adjust to current concept and preserve old knowledge.
Drift detection	Thoroughly studied	Can be used for task detection Some recent OCL work: [Gunasekara *et al.*, 2022a]. [Gunasekara *et al.*, 2022b].
Drift prediction.	Used when dealing with recurrent concept drifts.	Can be used for task prediction. Some SL work: [Chen *et al.*, 2016], [Suárez-Cetrulo *et al.*, 2023]
Missing labels	Some methods have been proposed to tackle this [Gomes *et al.*, 2022].	Yet to be fully explored. Can employ some of the SL approaches discussed in [Gomes *et al.*, 2022].

Topic	SL	OCL
Recurrent concept drifts	Similar to OCL, without explicit learning objective to preserve old knowledge. For latest research refer to [Suárez-Cetrulo *et al.*, 2023].	SL concept pool maintenance techniques [Suárez-Cetrulo *et al.*, 2023] can be useful in maintaining references to different NN structures in OCL parameter-isolation methods. Concept equivalence and concept similarity can be used to retrieve relevant instances or NN structures. Many more techniques are discussed in [Suárez-Cetrulo *et al.*, 2023].
Evaluation	Frameworks can employ OCL dual learning objective and metrics discussed in section 3.2. So SL methods and techniques can be evaluated under OCL setting.	Employs dual learning objective.
Application	Suitable for applications which needs to adjust to the current concept very quickly.	Suitable for applications which needs to adapt to current concept very quickly while preserving old knowledge.

But wait, there's more

- Data stream ML: change/drift detection and recovery
- Online Continual Learning: combatting forgetting
- Time series prediction: trends and seasonal behaviour
- Reinforcement learning: exploration/exploitation trade-off, probabilistic risk taking

Thank you

Some references:

Yaqian Zhang, Bernhard Pfahringer, Eibe Frank, Albert Bifet, Nick Jin Sean Lim, Yunzhe Jia:
A simple but strong baseline for online continual learning: Repeated Augmented Rehearsal. NeurIPS 2022

Nuwan Gunasekara, Bernhard Pfahringer, Heitor Murilo Gomes, Albert Bifet:
Survey on Online Streaming Continual Learning. IJCAI 2023

Anton Lee, Yaqian Zhang, Heitor Murilo Gomes, Albert Bifet, Bernhard Pfahringer:
Look At Me, No Replay! SurpriseNet: Anomaly Detection Inspired Class Incremental Learning. CIKM2023