Nieustannie uczące się systemy

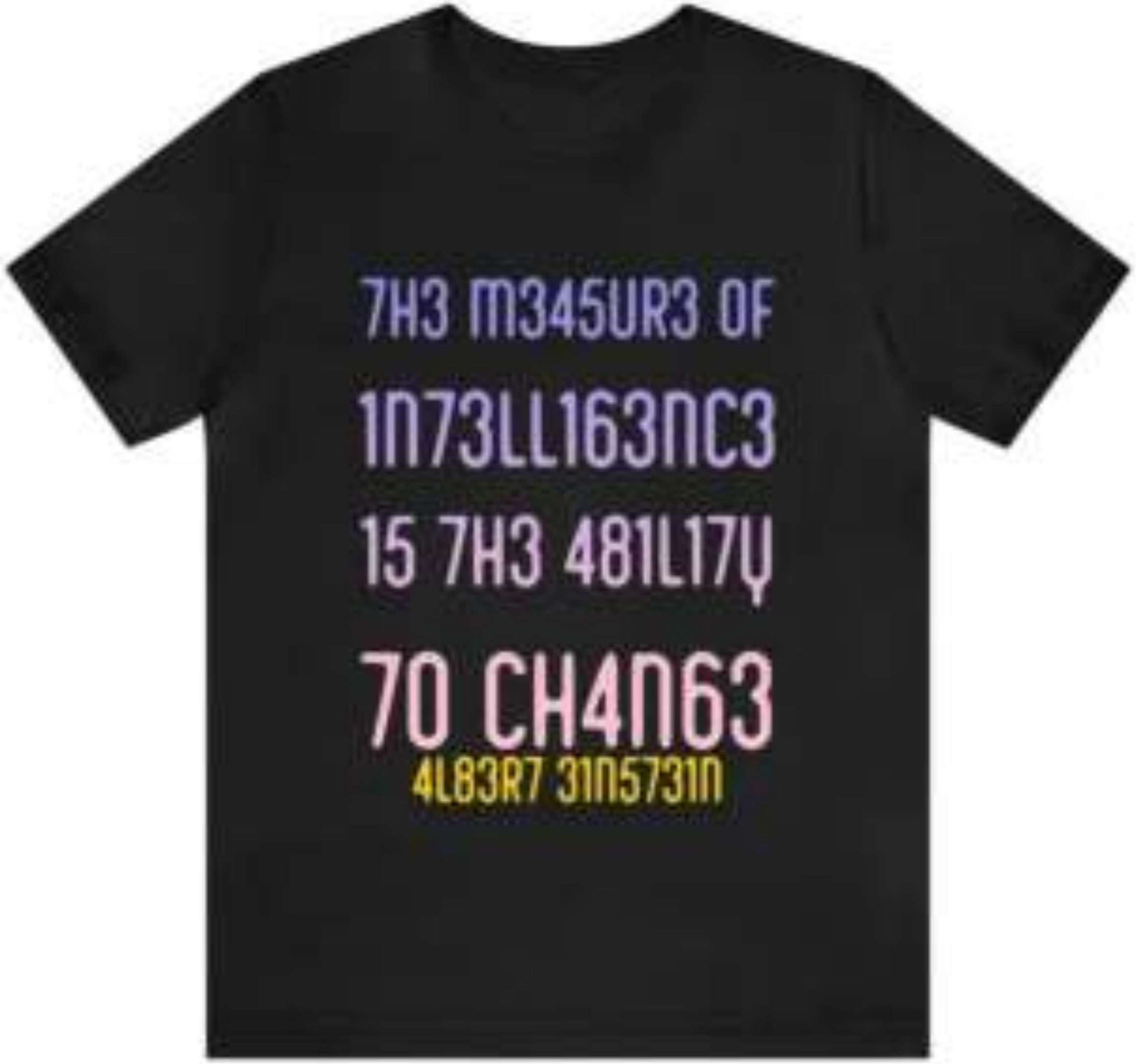
Michał Woźniak

Katedra Systemów i Sieci Komputerowych, Politechnika Wrocławska

E-mail: Michal.Wozniak@pwr.edu.pl https://www.kssk.pwr.edu.pl/users/mwozniak

Politechnika Wrocławska

KSSK


7H3 M345UR3 OF
1N73LL163NC3
15 7H3 481L17Y
70 CH4N63
4L83R7 31N5731N

Content of the talk

- Motivations of CL
- Continual learning scenarios
- Gaps
- Lifelong learning
- Selected methods
- Few ideas from my team
- Challenges and Conclusion

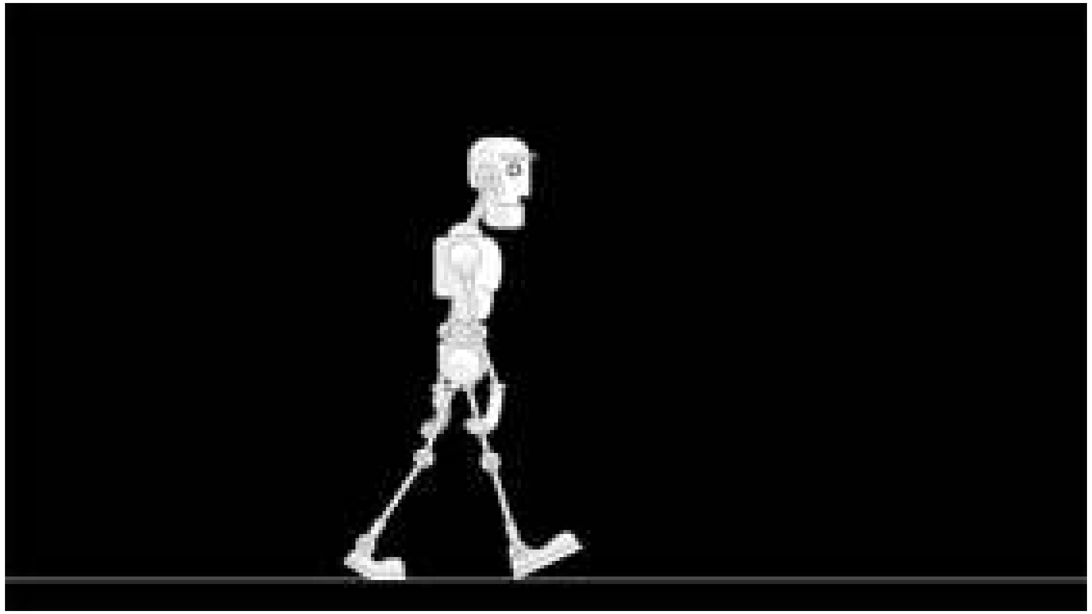


Illustration by David Parkins

Motivations

- Data is in motion
- We need working models asap
- Limited resources
- Possible probabilty distribution shift
- New tasks (or new classes) should be learned

Bing Liu testował na drodze autonomiczny samochód, gdy nagle coś poszło nie tak. Pojazd działał płynnie, aż dotarł do skrzyżowania i się zatrzymał. Liu i inni pasażerowie byli zaskoczeni. Droga, na której się znajdowali, była pusta, bez pieszych i innych samochodów w zasięgu wzroku.
"Rozejrzeliśmy się, nie zauważyliśmy nic niepokojącego" - mówi Liu, informatyk z Uniwersytetu Illinois w Chicago. Zdezorientowani inżynierowie przejęli kontrolę nad pojazdem i wrócili do laboratorium, aby przeanalizować podróż. Ustalili, że samochód został zatrzymany przez kamyk na drodze. Nie było to coś, co człowiek by zauważył, ale kiedy zostało zarejestrowane przez samochód jako nieznany obiekt - coś, z czym system sztucznej inteligencji (AI) prowadzący samochód nie spotkał się wcześniej.
Problem nie dotyczył algorytmu sztucznej inteligencji jako takiego - działał on zgodnie z protokołem bezpieczeństwa, zatrzymując się przed nieznanym obiektem. Problem polegał na tym, że gdy po zakończeniu uczenia systemu, który potrafił rozróżniać między czystą drogą a przeszkodą, system ten nie mógł nauczyć się niczego więcej. Kiedy napotkał coś, co nie było częścią jego danych treningowych, takich jak kamyk lub nawet ciemna plama na drodze, nie wiedział, jak zareagować.

Ludzie mogą opierać się na tym, czego się nauczyli i dostosowywać się do zmian zachodzących w ich otoczeniu; większość systemów AI obraca się w zakresie tego, co już wie.

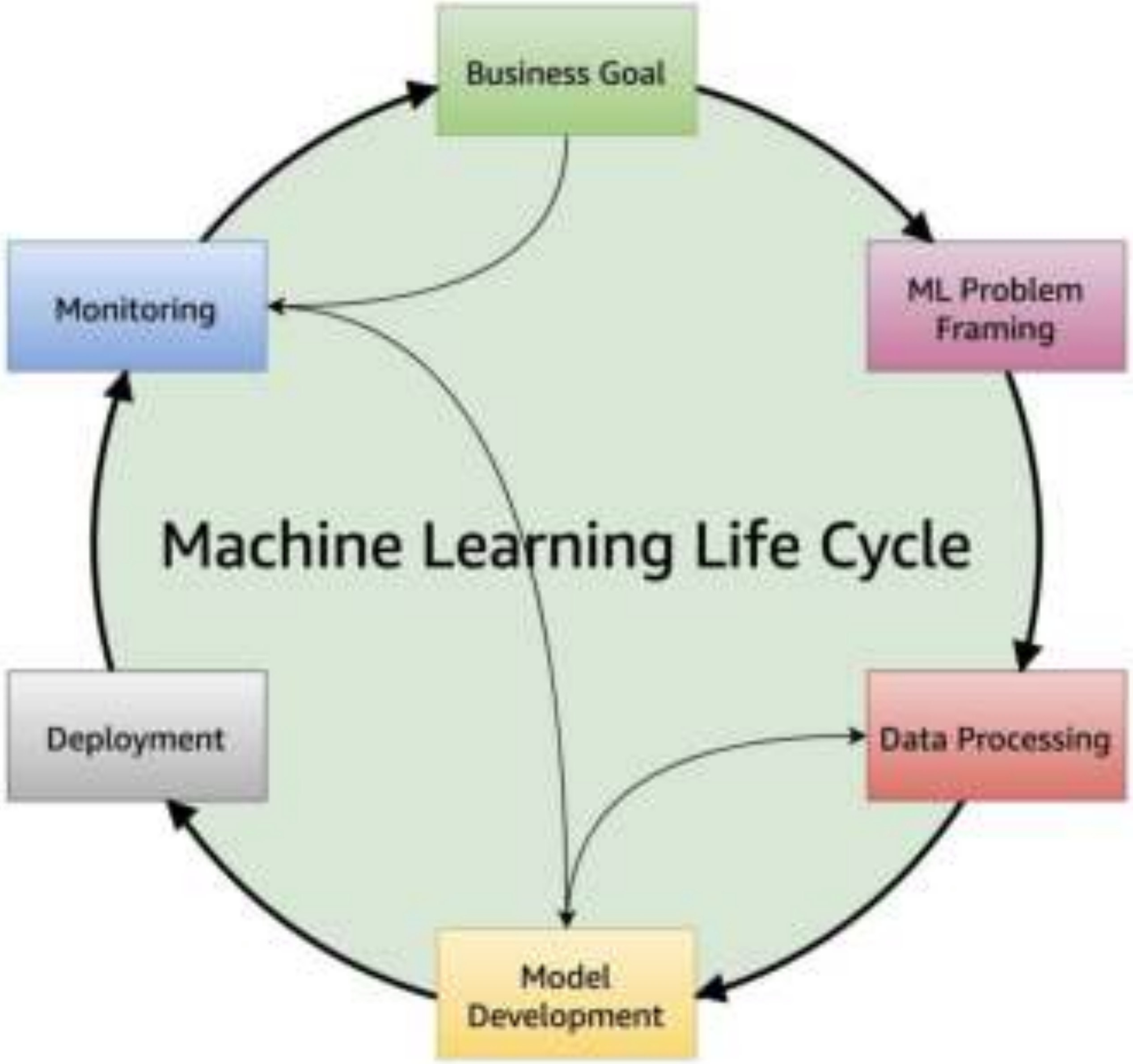
Neil Savage, *Learning over a lifetime,* https://www.nature.com/articles/d41586-022-01962-y, Nature, 20 July 2022

How can we improve AI efficiency, scalability and adaptability?

Hence making it sustainable in the long term

Vincenzo Lomonaco, University of Pisa & ContinualAI

Model monitoring system ensures your model is maintaining a desired level of performance.



Typical ML lifecycle

https://docs.aws.amazon.com/wellarchitected/latest/machine-learning-lens/well-architected-machine-learning-lifecycle.html

Possible scenarios

- The typical machine learning task is **isolated learning** because it does not consider any other related information or the previously learned knowledge.

- The learning environments are typically static and closed.

- Human never learns in isolation or from scratch. Human always retains the knowledge learned in the past and uses it to help future learning and problem solving.

- Online learning (stationary data stream)
- Learning from nonstationary data streams (concept drift), somethimes can be considered as domain adaptation

Model for the same task

- Task incremental learning
- Domain incremental learning
- Class incremental learning

Model for the diffrent tasks

- Online learning (stationary data stream)
- Learning from nonstationary data streams (concept drift), somethimes can be considered as domain adaptation

Model for the same task

- Task incremental learning
- Domain incremental learning
- Class incremental learning

Model for the diffrent tasks

Online learning (stationary data streams)

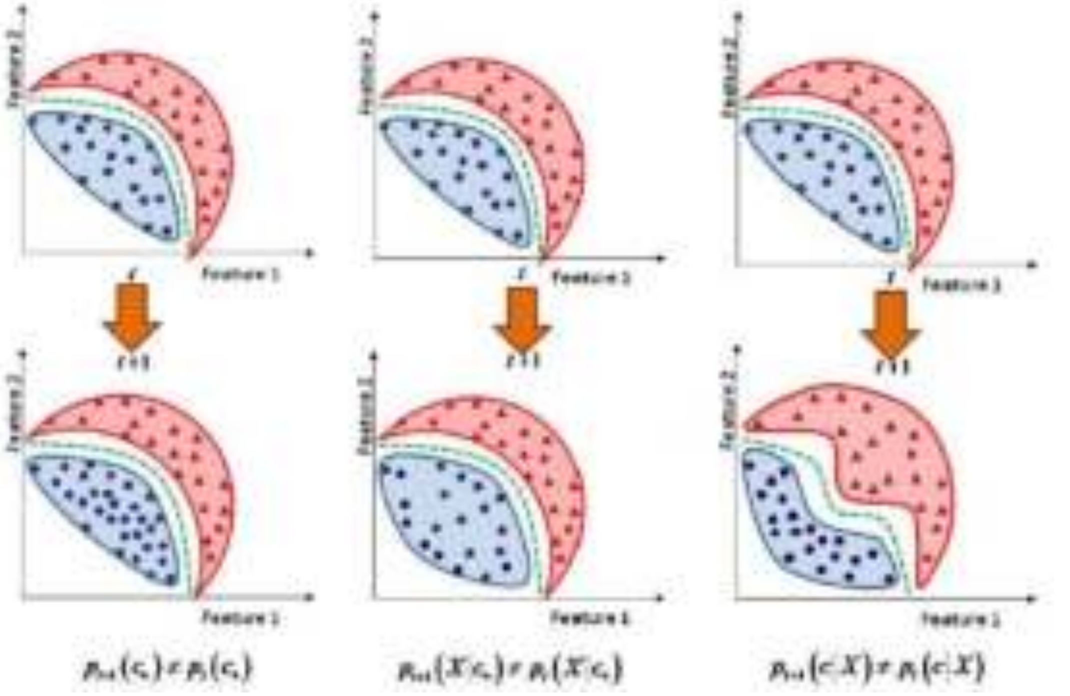
- The data distribution is stationary.
- The goal is to build a model as quickly as possible to use it and simultaneously improve it when new data arrives.
- This group relates to the family of algorithms that continuously update the classifier parameters while processing the incoming data.
    - Each object must be processed at least once in the course of training.
    - Limited resources.
    - The training process can be paused at any time, and its accuracy should not be lower than that of a classifier trained on batch data collected up to the given time.

P. Domingos, G. Hulten, A general framework for mining massive data streams, Journal of Computational and Graphical Statistics 12, 2003, 945–949.

Frank
Sinatra
(When I Was Seventeen)
It Was A
Very Good
Year
From The Reprise Album
September Of My Years
reprise

Learning from nonstationary data streams (concept drift), sometimes can be considered as domain adaptation

- The data distribution is nonstationary.
- The goal is to build a model using data from current distribution and update it taking into consideration that concept drift may appear.
- Forgetting is our friend in this case☺


Feature 2
Feature 1
t
t+1
$p_{t+1}(c_i) \neq p_t(c_i)$
$p_{t+1}(X|c_i) \neq p_t(X|c_i)$
$p_{t+1}(c|X) \neq p_t(c|X)$

Krawczyk B. et al., Ensemble learning for data stream analysis: A survey, Information Fusion, 2017

Should we react to the virtual concept drift

- Olivera et al. noted that although virtual concept drift does not affect the change in decision boundaries and has not been the focus of much research, it is important to note that it can also affect the usefulness of learned decision boundaries by classifiers, e.g., for unrepresentative learning sets used to build a classifier.

- Therefore, the maintained models need to be updated regardless of the occurring concept drift type.

G. Oliveira et al., Tackling virtual and real concept drifts: An adaptive gaussian mixture model approach, 2021.

If we require the learner to master new knowledge (be able to solve a new task), can we afford to train the model from scratch?

Shouldn't we look for learning methods that allow us to add new knowledge to solve a new task on the one hand, but also not to forget what the model has learned so far?

Just how much does it cost to train a model?

Costs of training differently sized BERT models on the Wikipedia and Book corpora (15 GB):

GPT3 175 bilion parameters.
GPT4 is based on eight models with 220 billion parameters each, for a total of about 1.76 trillion parameters, connected by a Mixture of Experts (MoE).

$2.5k - $50k (110 million parameter model)

$10k - $200k (340 million parameter model)

$80k - $1.6m (1.5 billion parameter model)

Training costs can vary drastically due to different technical parameters, climbing up to US$1.3 million for a single run when training Google's 11 billion parameter Text-to-Text Transfer Transformer (T5) neural network model variant. A project that might require several runs could see total training costs hit a jaw-dropping US$10 million.

Or Sharir et al., The cost of training NLP models a concise overview, https://arxiv.org/pdf/2004.08900.pdf

We are not looking for incremental improvements in state-of-the-art AI and neural networks, but rather paradigm-changing approaches to machine learning that will enable systems to continuously improve based on experience.

Hava Siegelmann, 2018

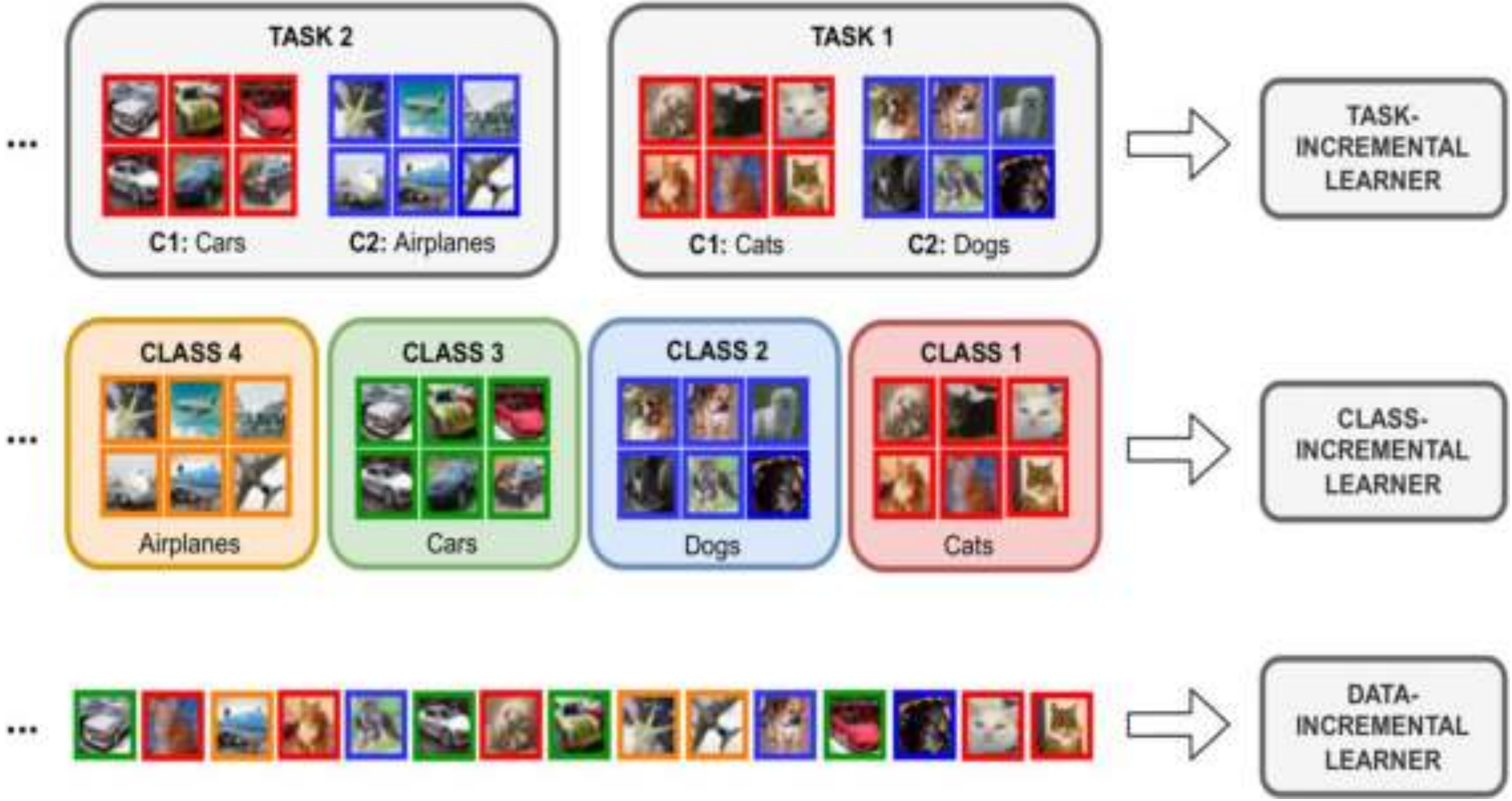
Professor of Computer Science, and Brain Sciences, University of Massachusetts, Amherst

- Online learning (stationary data stream)
- Learning from nonstationary data streams (concept drift), somethimes can be considered as domain adaptation

Model for the same task

- Task incremental learning
- Domain incremental learning
- Class incremental learning

Model for the diffrent tasks

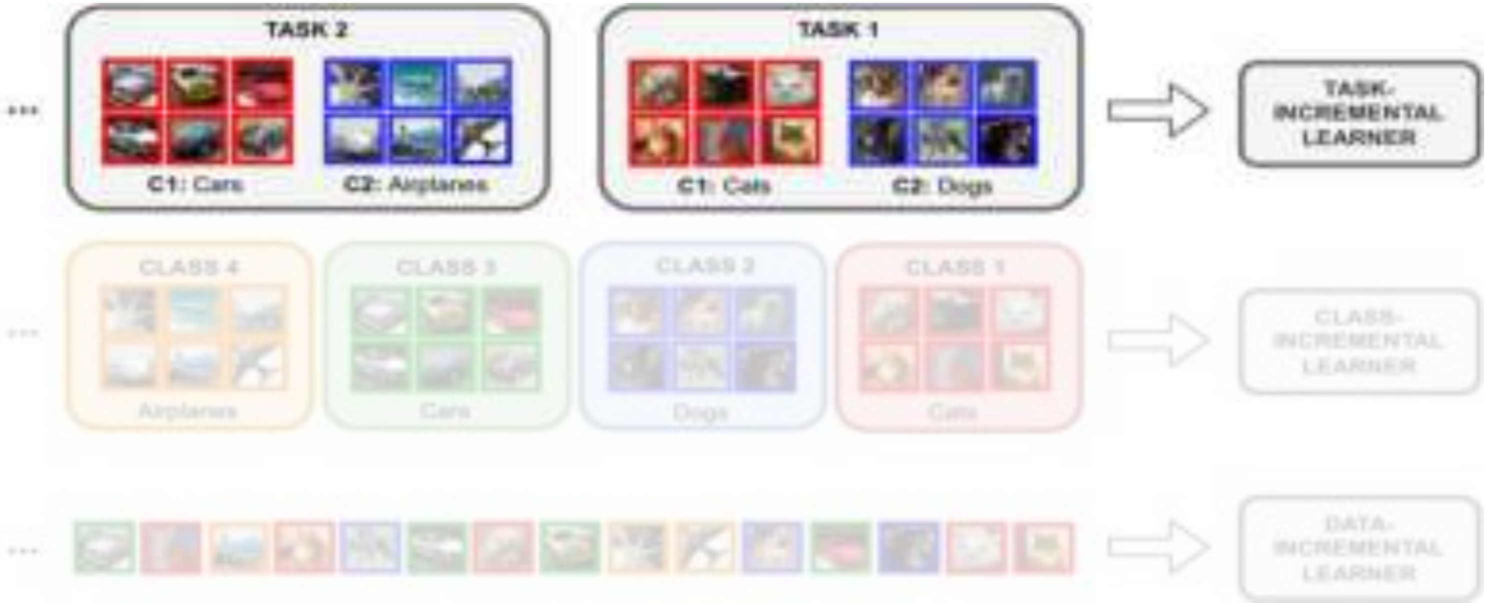


Example given by Bartosz Krawczyk, RIT

Task incremental learning

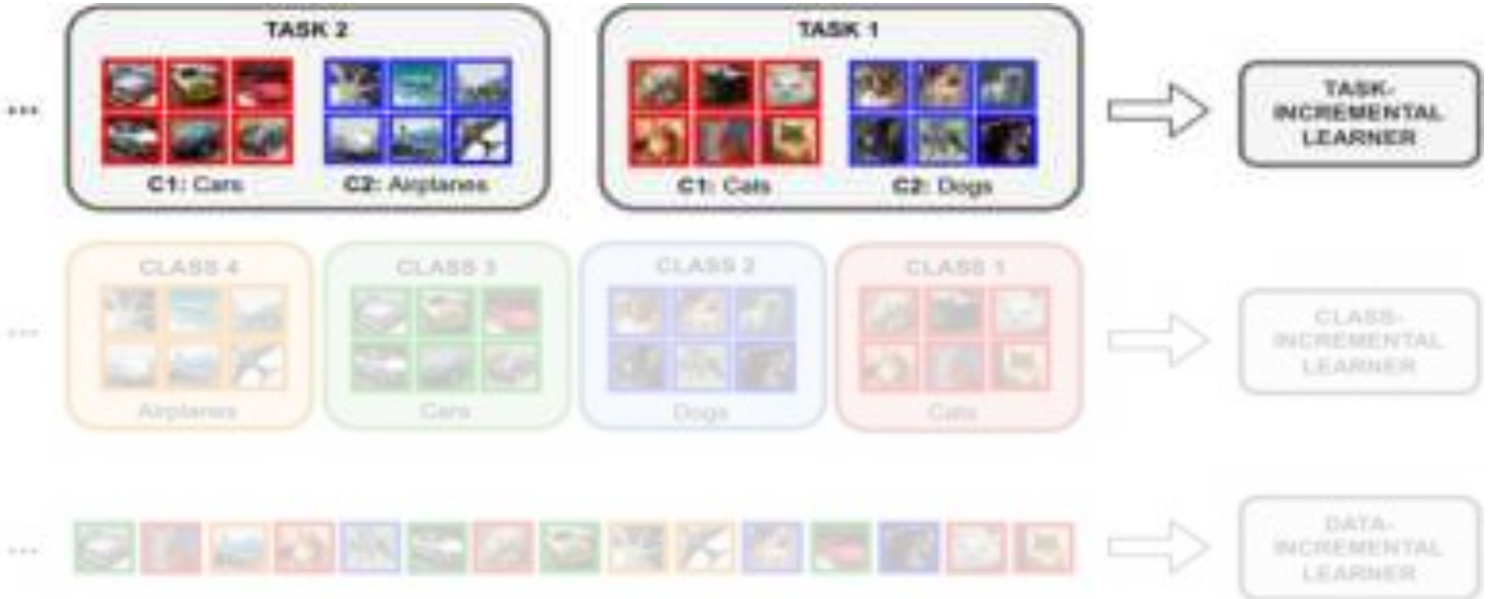
- Models are always informed about which task needs to be performed.
- Since task identity is always provided, i.e., we train models with task-specific components.
- A typical network architecture used in this scenario has a “multi-headed” output layer, meaning that each task has its own output units but the rest of the network is (potentially) shared among tasks.

Gido M. van de Ven and Andreas S. Tolias, Three scenarios for continual learning, https://arxiv.org/abs/1904.07734



Domain incremental learning

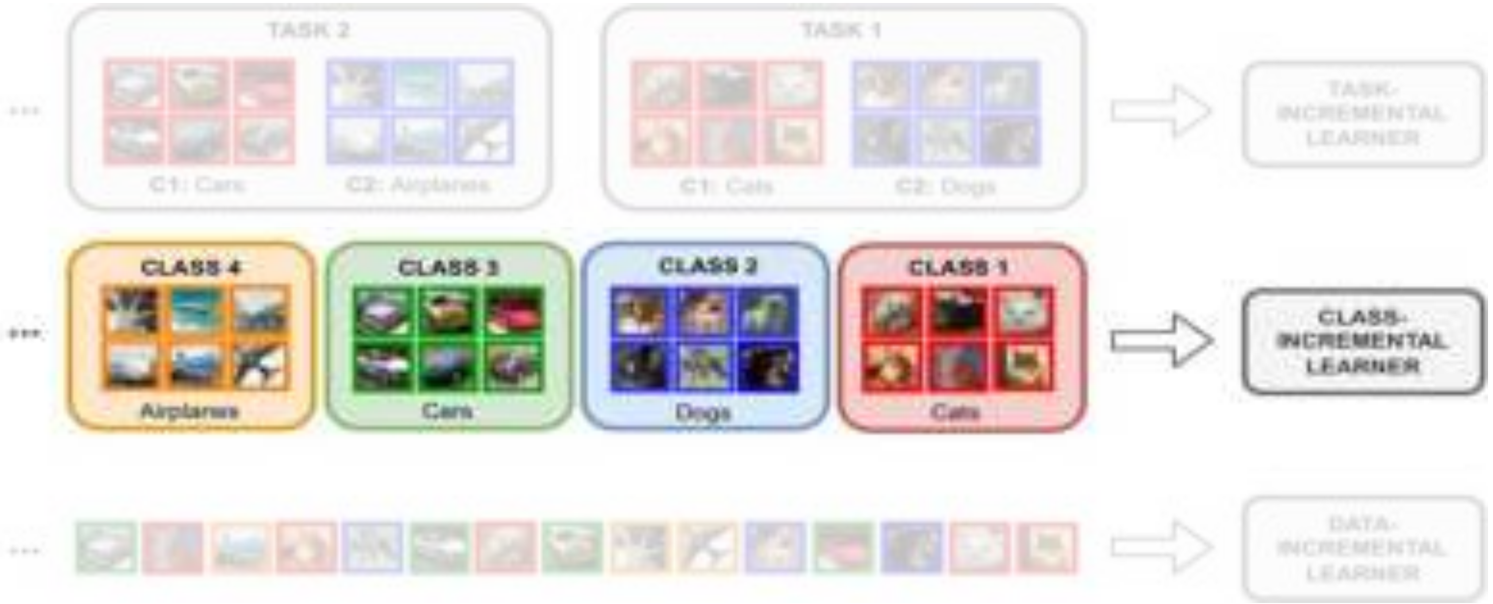
- The task identity is not available at test time.
- Models however only need to solve the task at hand; they are not required to infer which task it is.
- A relevant real-world example is an agent who needs to learn to survive in different environments, without the need to explicitly identify the environment it is confronted with.



Class incremental learning

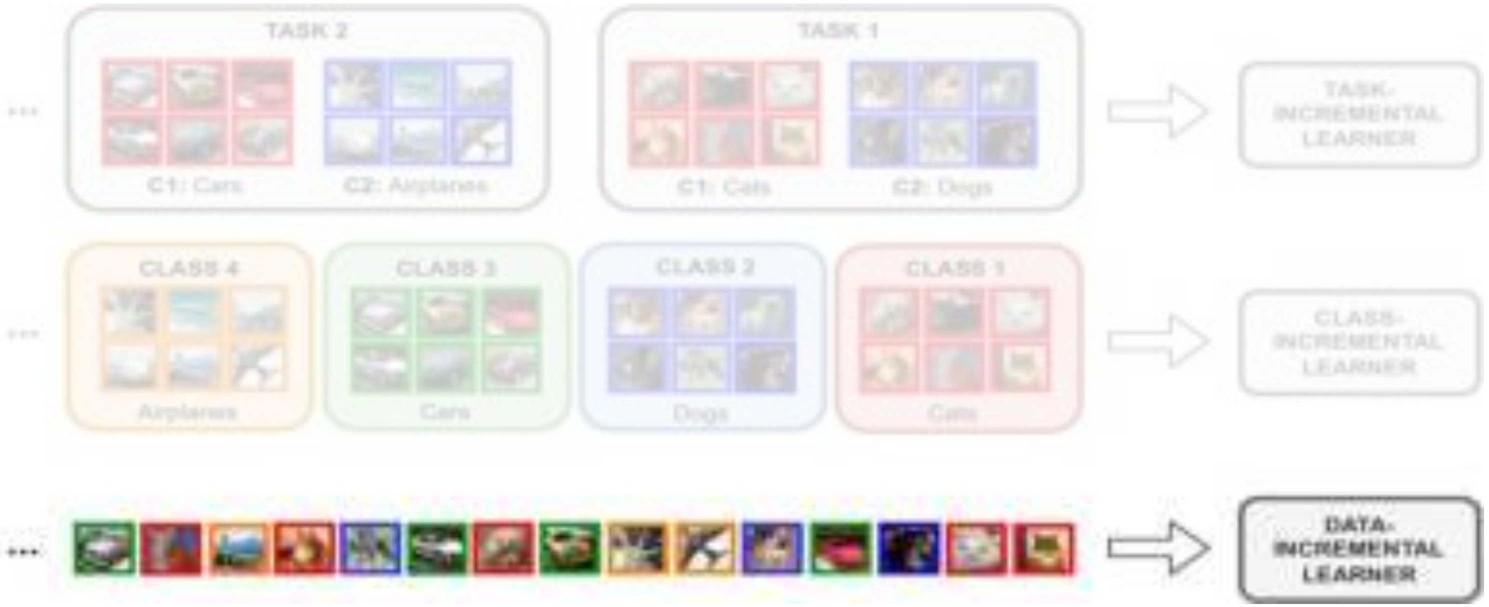
Models must be able to both solve each task seen so far and infer which task they are presented with.

It includes the common real-world problem of incrementally learning new classes of objects.



Data incremental learning

Each block is a batch of data



- Online learning (stationary data stream)
- Learning from nonstationary data streams (concept drift),

Model for the
same task

WHAT ARE THE GAPS?

- Domain incremental learning
- Class incremental learning

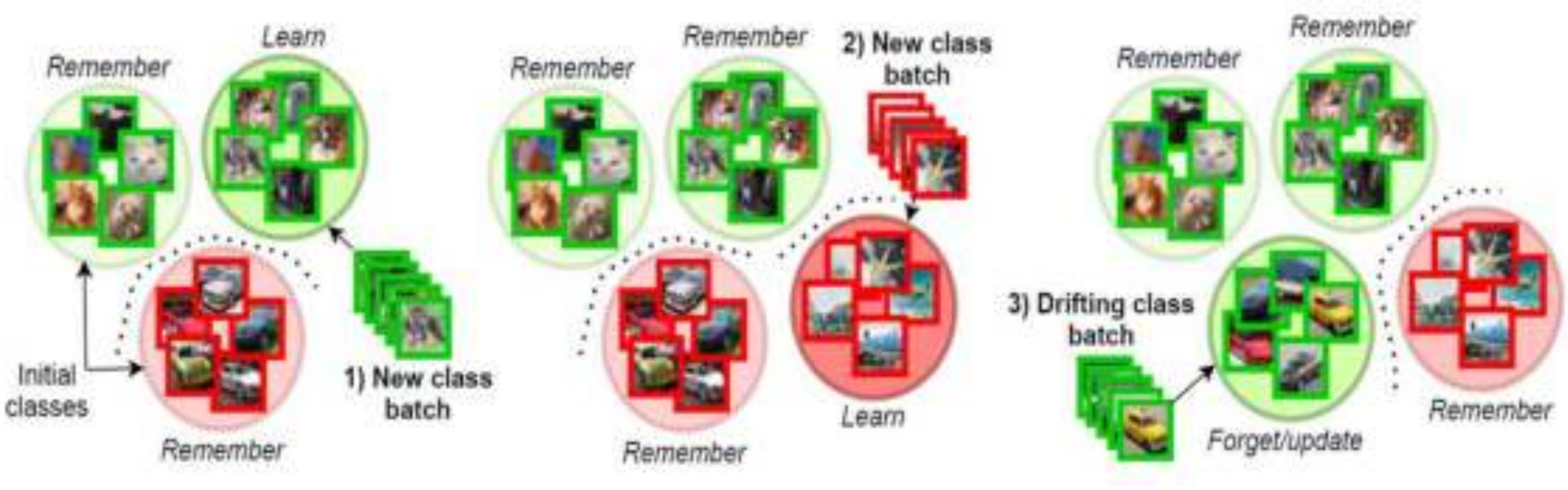
Model for the
diffrent tasks

Continual learning methods for stationary data (task/class incremental learning) focus on knowledge retention.

They tackle catastrophic forgetting and are unable to efficiently handle concept drifts.

Algorithms for non-stationary data are designed around change adaptation.

They focus on concept drifts. But do not consider catastrophic forgetting explicitly.



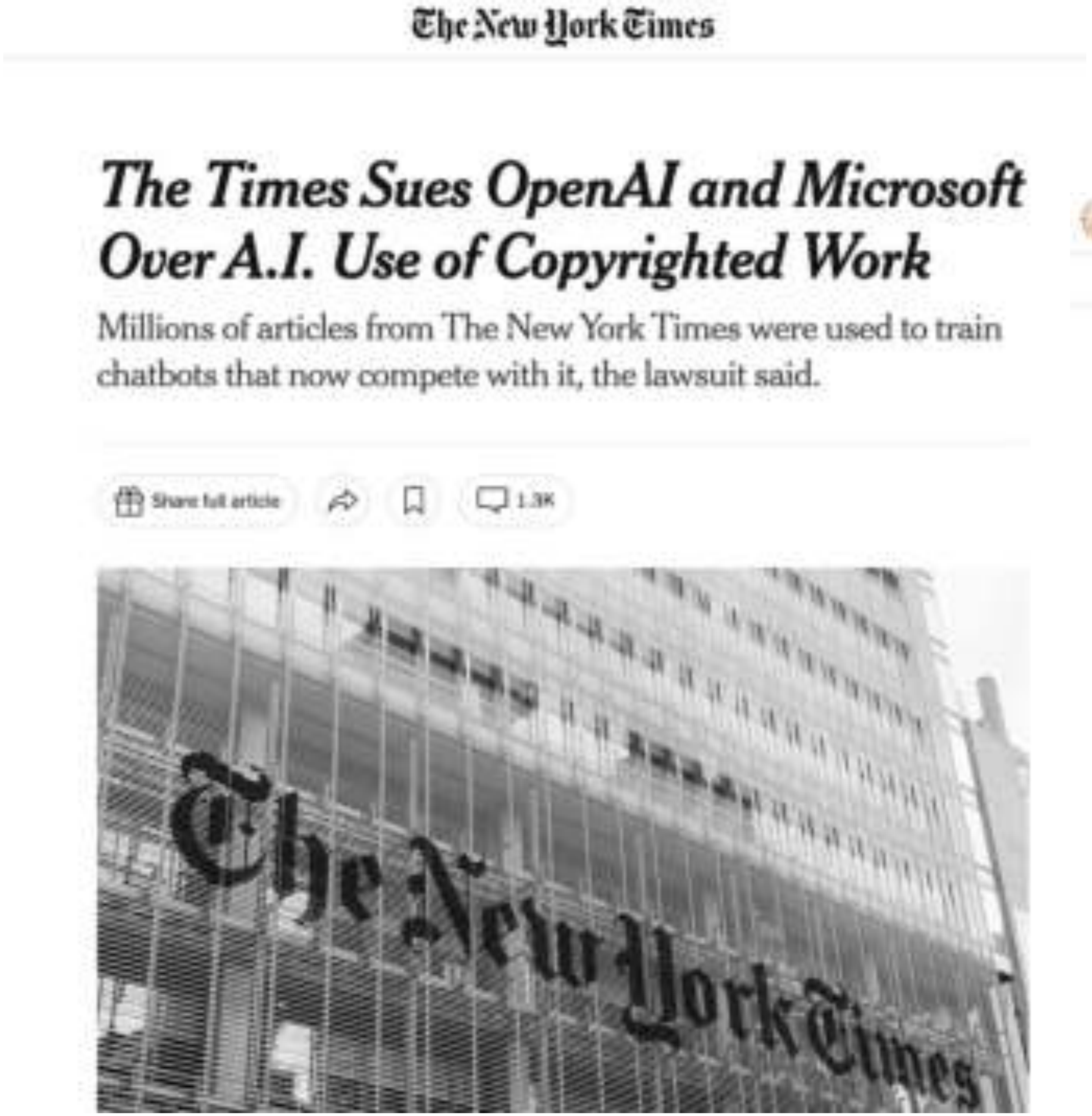
Learning new classes, retaining previous knowledge and adapting to concept drifts, illustrated by the example of a binary recommendation system.

The lack of holistic solutions!

Korycki L., Krawczyk B., Class-Incremental Experience Replay for Continual Learning under Concept Drift, CVPR 2021

The described scenarios assume that the data arrives in a sequential manner, and we upgrade model using incoming examples.

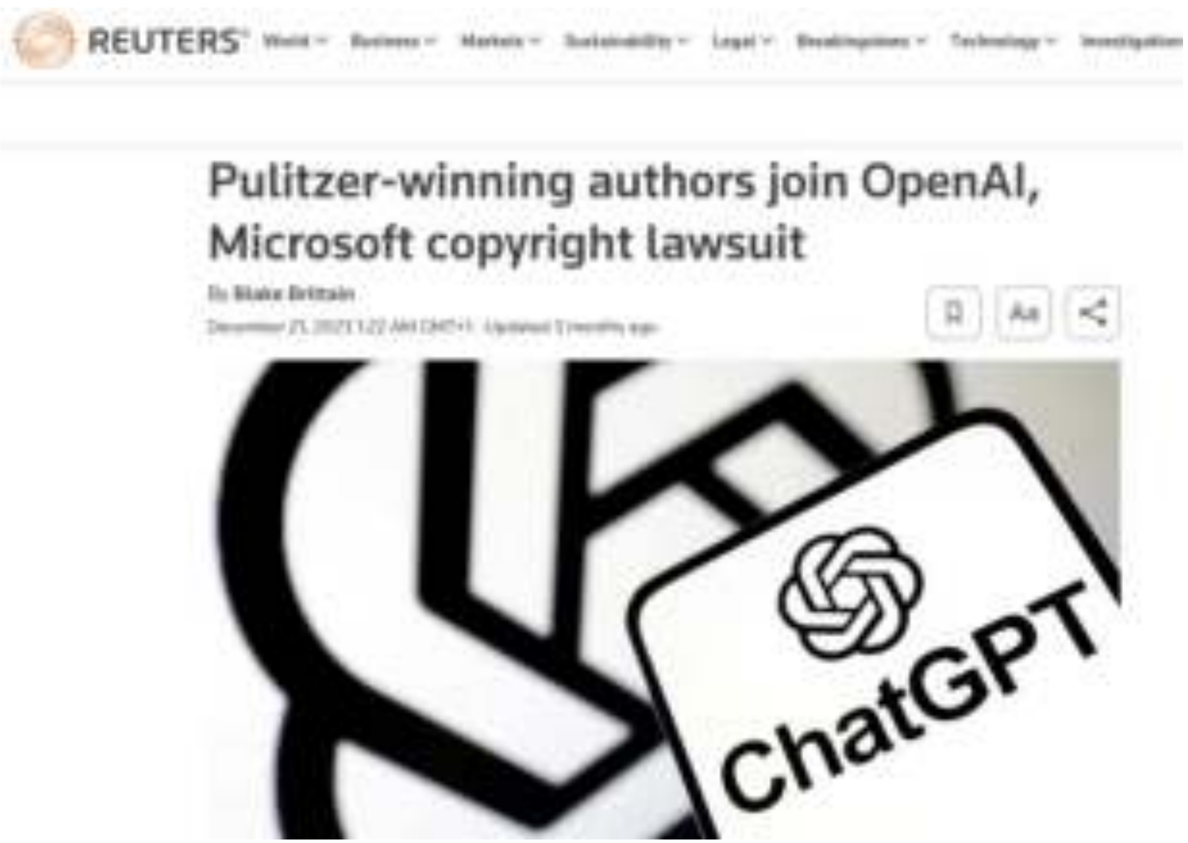
But the **data may be corrected, or deleted**.



The New York Times

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.



Pulitzer-winning authors join OpenAI, Microsoft copyright lawsuit

By Blake Brittain

December 21, 2023 1:22 AM GMT+1 · Updated 3 months ago

Motivations:

- privacy (e.g., GDPR -> right to be forgotten),
- fairness (e.g., bias removal),
- continual learning (data upgrading or removal)

How to deal with this scenario?

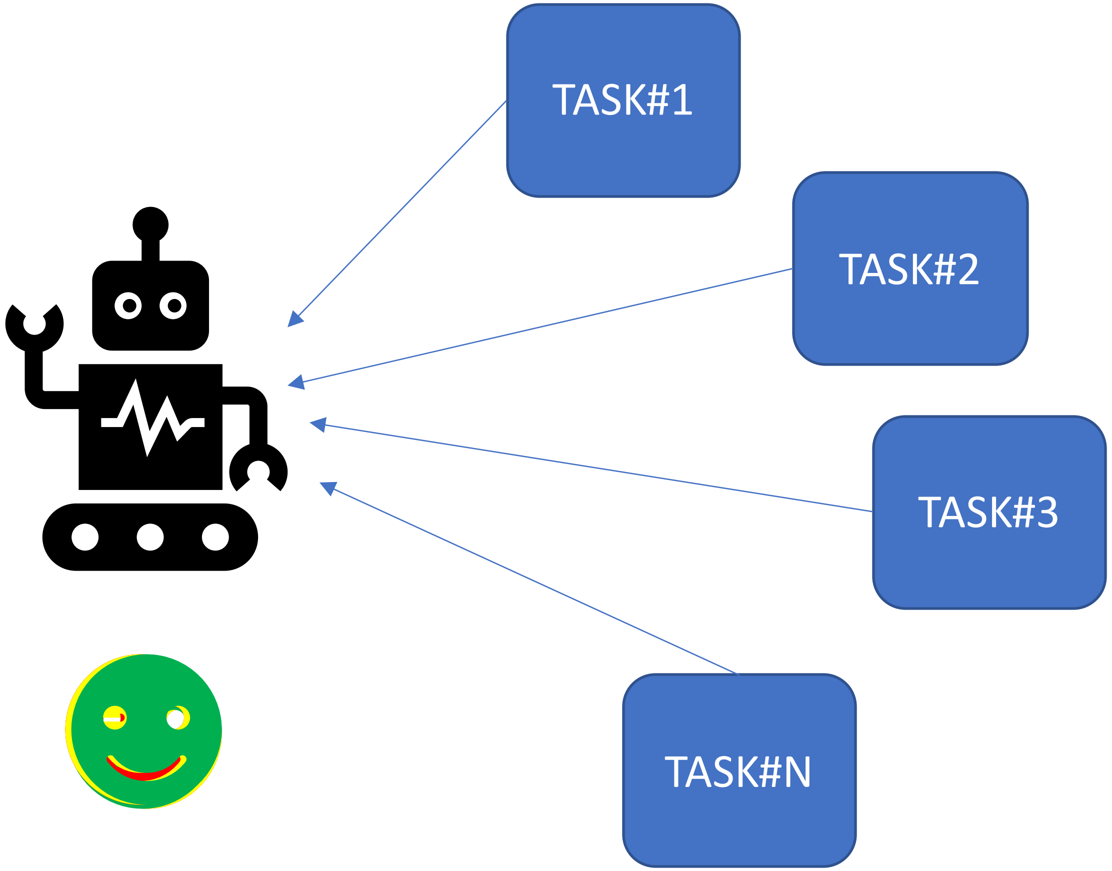
Machine unlearning

Removing the influence of a specified subset of training data from a machine learning model is an important challenge.

Yinzhi Cao and Junfeng Yang (2015), Towards Making Systems Forget with Machine Unlearning, IEEE Symposium on Security and Privacy
Lucas Bourtoule et al. (2021), Machine Unlearning, IEEE Symposium on Security and Privacy
Mercuri S. et al., An Introduction to Machine Unlearning, https://arxiv.org/abs/2209.00939

Lifelong ML definition and objectives

Lifelong machine learning aims to imitate the human learning process and capability



- Lifelong ML learns continually, retains the knowledge learned in the past, and uses the accumulated knowledge to help future learning and problem solving.

- A large number of labeled training examples is not necessary.

- In many domains, no training data is needed at all because they may already be covered by some other/past domains and such similar past domains can be automatically discovered.

Lifelong ML is a continuous learning process.

At any point in time, the learner has performed a sequence of N tasks, T1, T2, … , TN .

These (previous) tasks have their corresponding datasets D1, D2, …, DN .

The tasks can be of different types and from different domains.

When faced with the TN+1 (new or current) task with its data DN+1, the learner can leverage the past knowledge in the knowledge base (KB) to help learn TN+1 .

The task may be given or discovered by the system itself.

The objective of LL is usually to optimize the performance of the new task TN+1, but it can optimize any task by treating the rest of the tasks as the previous tasks.

KB maintains the knowledge learned and accumulated from learning the previous tasks.

Ideally, an lifelong learner should also be able to:

- learn and function in the open environment, where it not only can apply the learned model or knowledge to solve problems but also discover new tasks to be learned,

- learn to improve the model performance in the application or testing of the learned model.

The definition indicates five key characteristics of lifelong ML:

- continuous learning process,
- knowledge accumulation and maintenance,
- the ability to use the accumulated past knowledge to help future learning,
- the ability to discover new tasks,
- the ability to learn while working or to learn on the job.

Additionally, we have to consider:

- No access (or very limited) to previously encountered data.
- Limited resources (computational and memory resources)
- Incremental development of ever more complex knowledge and skills

Independent tasks

Each task can be learned independently, although due to their similarities and sharing of some latent structures or knowledge, learning Ti can leverage the knowledge gained from learning previous tasks.

Dependent tasks

Each task Ti has some dependence on some other tasks, e.g., in open-world learning (class incremental learning) each new supervised learning task adds a new class to the previous classification problem, and needs to build a new multi-class classifier that is able to classify data from all previous and the current classes.

- The tasks do not have to be from the same domain.

- The shift to the new task can happen abruptly or gradually, and the tasks and their data do not have to be provided by some external systems or human users.

Techniques we may use

- **Multi-task learning (MTL)** learns multiple related tasks simultaneously, aiming at achieving a better performance by using the relevant information shared by multiple tasks

$$\sum_{t=1}^{N}\sum_{i=1}^{n_t}\mathcal{L}(f(x_i^t), y_i^t)$$

- **Online multi-task learning** aims to learn the tasks sequentially and accumulate knowledge over time and leverage the knowledge to help subsequent learn ing (or to improve some previous learning task).
- Online MTL is thus lifelong ML.

- **Transfer learning** usually involves two domains: a source domain and a target domain.
- The **goal of transfer learning** is to use the labeled data in the source domain to help learning in the target domain.
- Although there can be more than one source domain, in almost all existing research only one source domain is used.
- The source domain normally has a large amount of labeled training data while the target domain has little or no labeled training data.

Lifelong ML vs. Transfer Learning

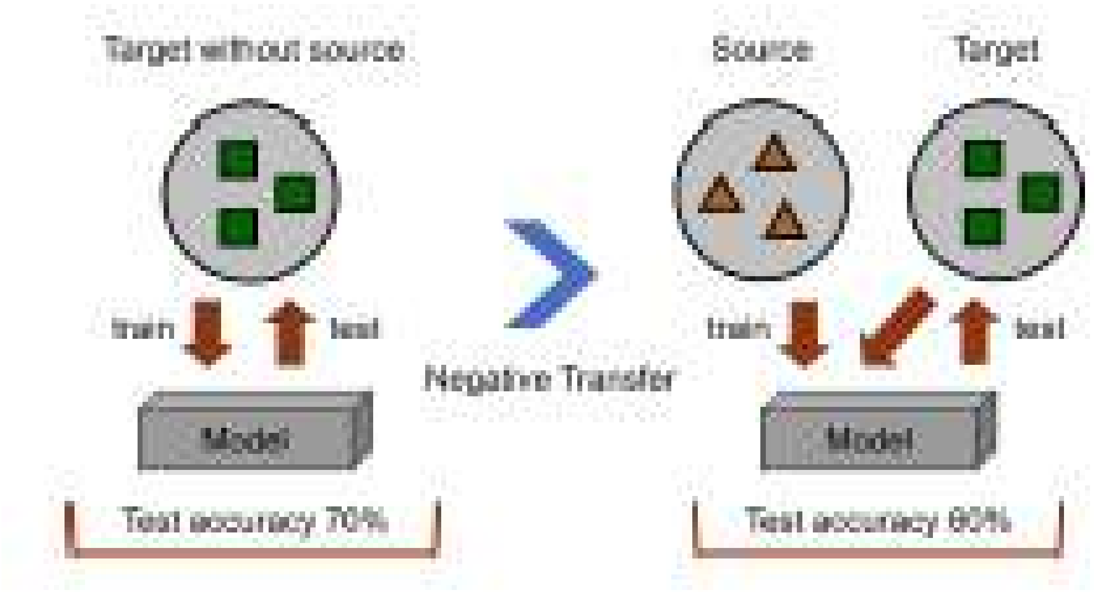
Transfer Learning: Learner can solve task 2 because it has learned task 1, but we do not care whether it can still do task 1.

Lifelong ML: Even though learner has learned task 2, but it does not forget task 1.

The effectiveness of TL is not always guaranteed, unless its basic assumptions are satisfied:

1) the learning tasks in the two domains are related/similar;
2) the source and target domain data distributions are not too different;
3) a suitable model can be applied to both domains.

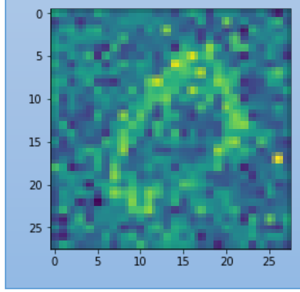
Violations of these assumptions may lead to **negative transfer** (NT), i.e., introducing source domain data/knowledge undesirably decreases the learning performance in the target domain



Wen Zhang et al., A Survey on Negative Transfer, IEEET NNLS, 2021

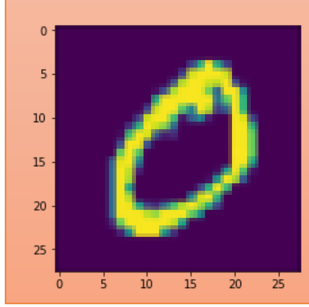
3 layers, 50 neurons each (example given by Hung-yi Lee, NTU)

Task 1

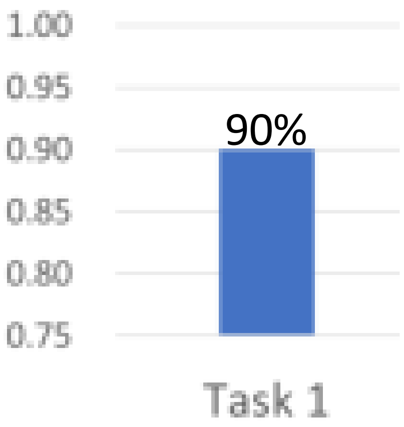


This is “0”.

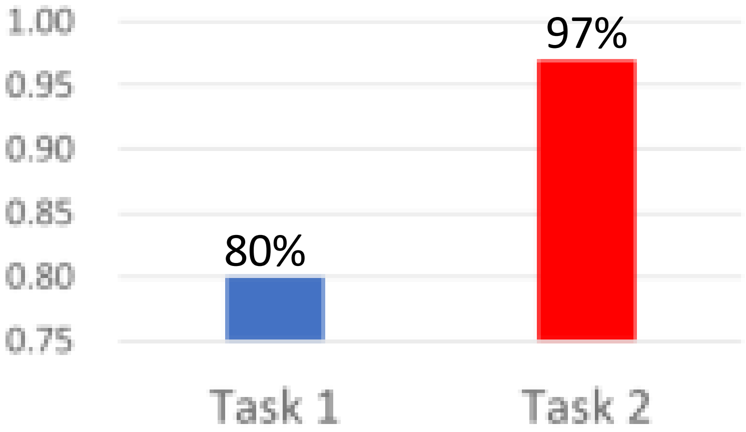
Task 2

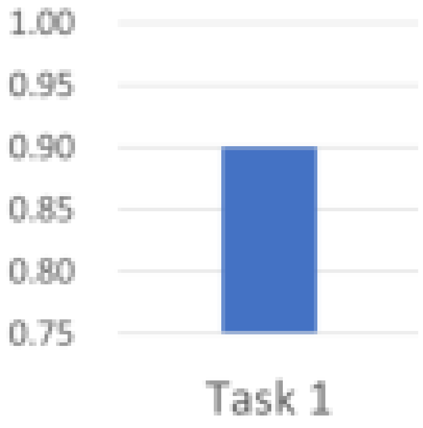


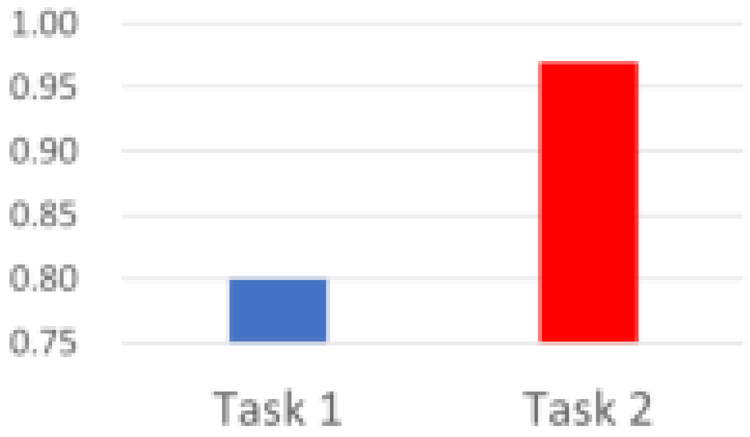
This is “0”.



What’s going on?



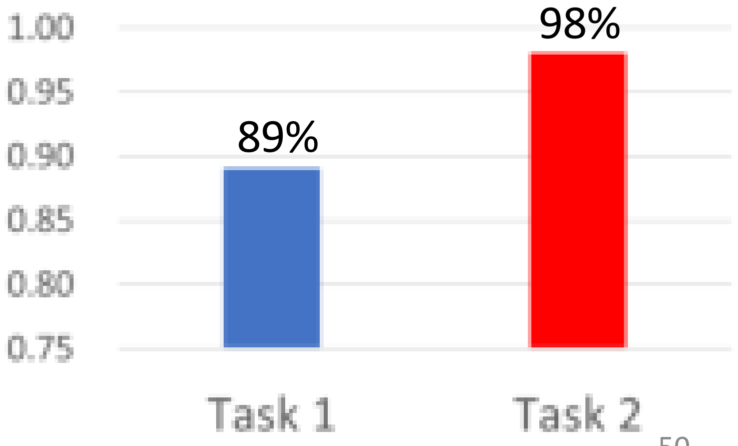




The network has enough capacity to learn both tasks.

Learning Task 1

Learning Task 2



- Using all the data for training – computational costly.
- Memorizing all data – breaking assumption on limited storage size.
- Multi-task training can be considered as the upper bound of lifelong ML.

Training model for each task – cannonical approach, which requires model for each taks.

Drawbacks:

- Limited memory for memorizing all the models
- Knowledge cannot transfer across different tasks

- Catastrophic forgetting or catastrophic interference was first recognized by McCloskey and Cohen in 1989

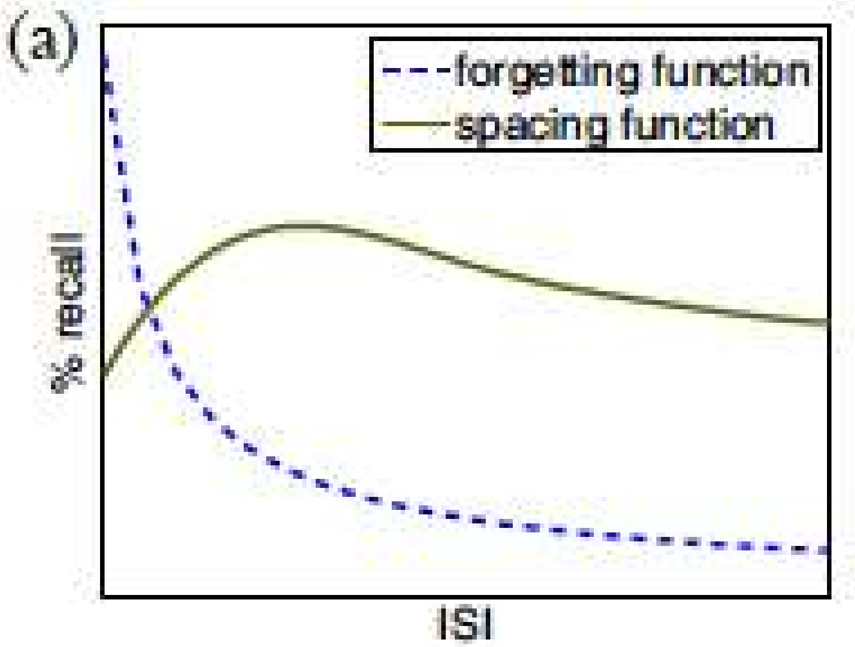
- They found that, when training on new tasks or categories, a neural network tends to forget the information learned in the previous trained tasks.

- Without fixing this problem, a single neural network will not be able to adapt itself to an lifelong ML scenario, because it forgets the existing information/knowledge when it learns new task.

- This was also referred to as the **stability-plasticity dilemma**.

- On the one hand, if a model is too stable, it will not be able to consume new information from the future training data.
- On the other hand, a model with sufficient plasticity suffers from large weight changes and forgets previously learned representation.

Is it typical for ML only – how it looks in humans?



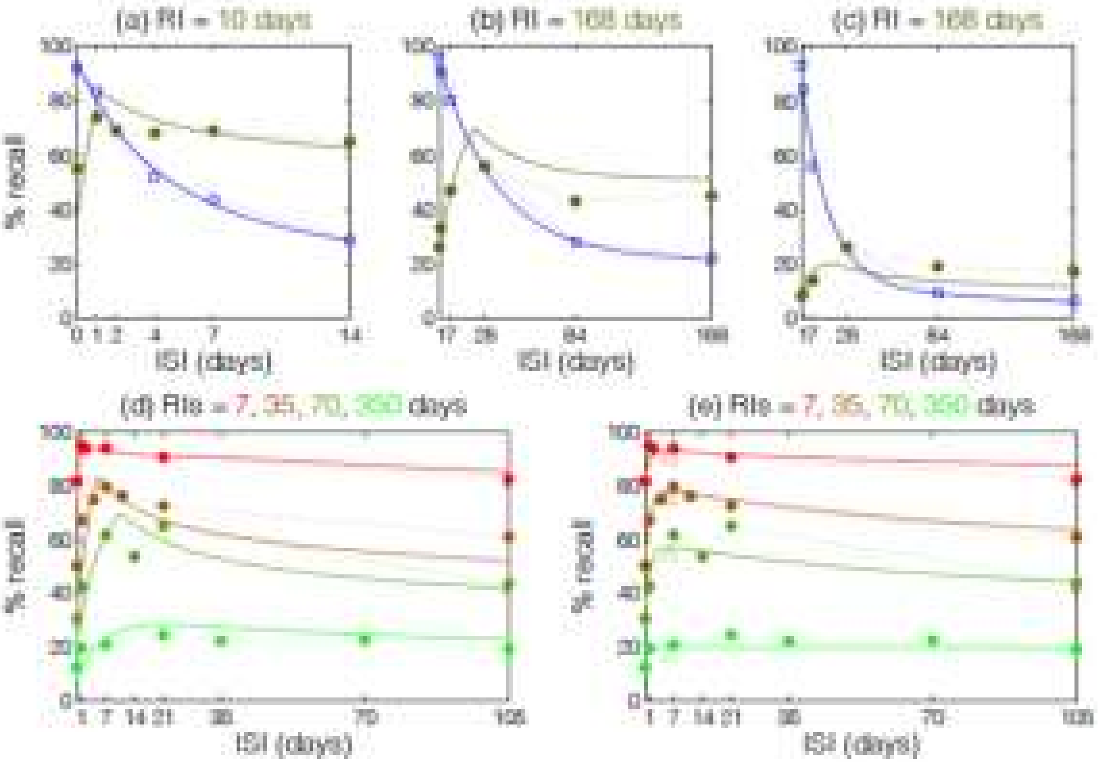


Figure 2: Modeling and experimental data of (Cepeda et al., in press) (a) Experiment 1 (Swahili-English), (b) Experiment 2a (obscure facts), and (c) Experiment 2b (object names). The four RI conditions of Cepeda et al. (2008) are modeled using (d) MCM and (e) the Bayesian multiscale model of Kording et al. (2007). In panel (e), the peaks of the model's spacing functions are indicated by the triangle pointers.

ISI (intersession interval) - time between training sessions

RI (retention time) – lag between session and test

Mozer M., C. et al., Predicting the Optimal Spacing of Study: A Multiscale Context Model of Memory, NIPS 2009

Li and Hoiem characterized three sets of parameters in a typical approach:

- $_s$: set of parameters shared across all tasks;
- $_o$: set of parameters learned specifically for previous tasks; and
- $_n$: randomly initialized task-specific parameters for new tasks.

Zhizhong Li, Derek Hoiem, Learning without Forgetting, IEEE T PAMI, 2017

Feature extraction

Both θ_s and θ_o remain the same while the outputs of some layers are used as features for training θ_n for a new task.

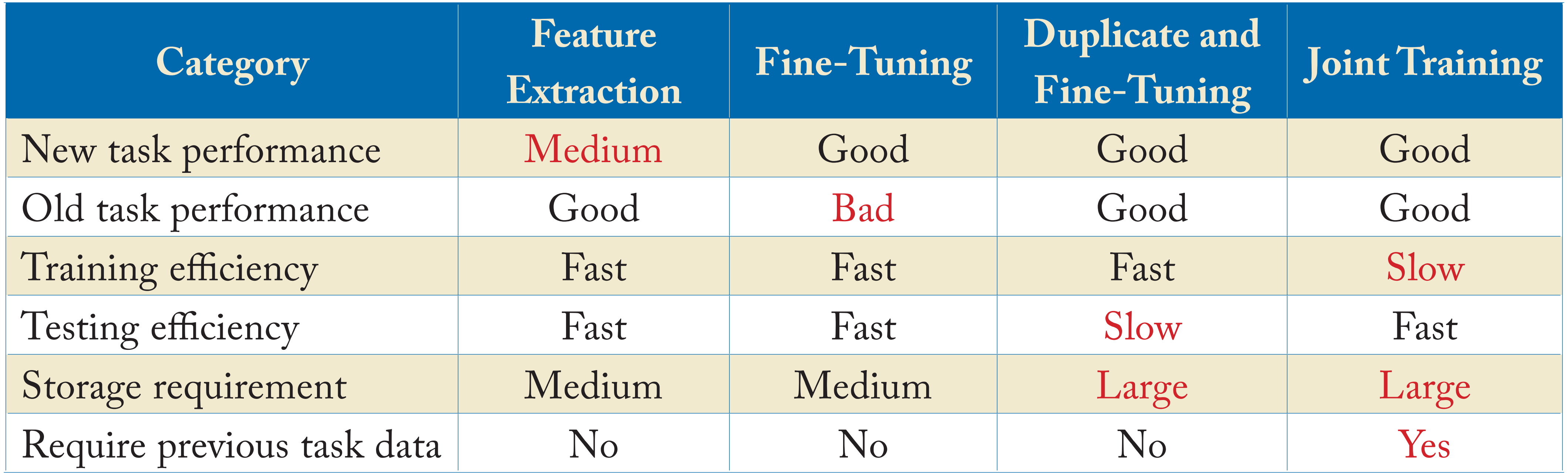
Fine-tuning

- θ_s and θ_n are optimized and updated for the new task while θ_o remain fixed.
- To prevent large shift in θ_s, a low learning rate is typically applied.
- Also, for the similar purpose, the network can be duplicated and fine-tuned for each new task, leading to networks for N tasks.
- Some authors propose to fine-tune parts of θ_s, e.g. top layers (a kind of the trade off between fine-tuning and feature extraction).

Join training:

- All parameters θ_s, θ_o, and θ_n are jointly optimized accross all tasks, that requires storing all training data of all tasks.
- Multi-taks learning typically takes this approach.

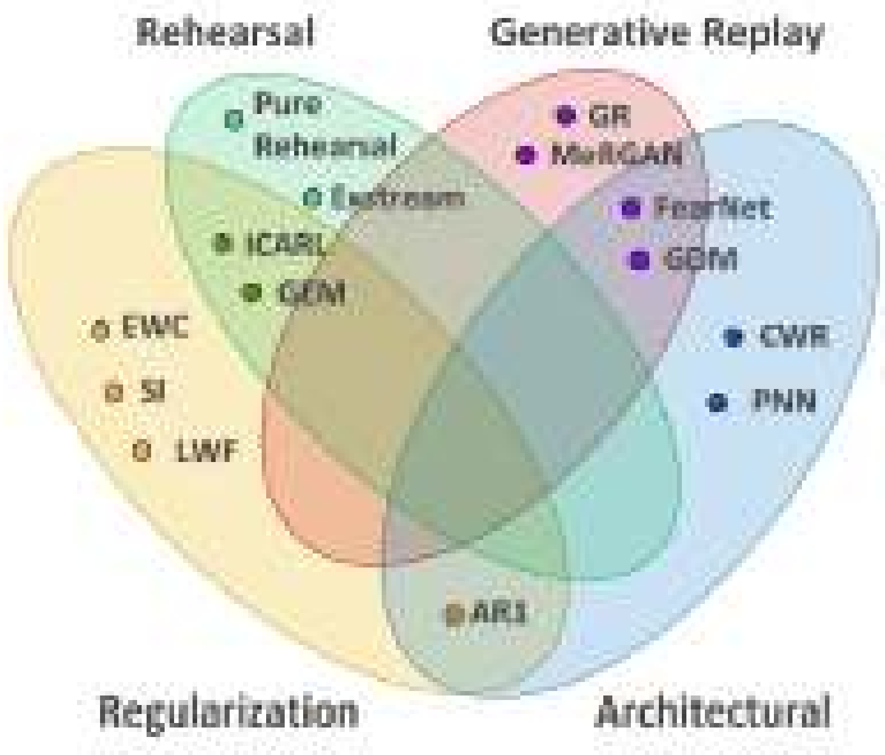
Li and Hoiem also presented the following summary

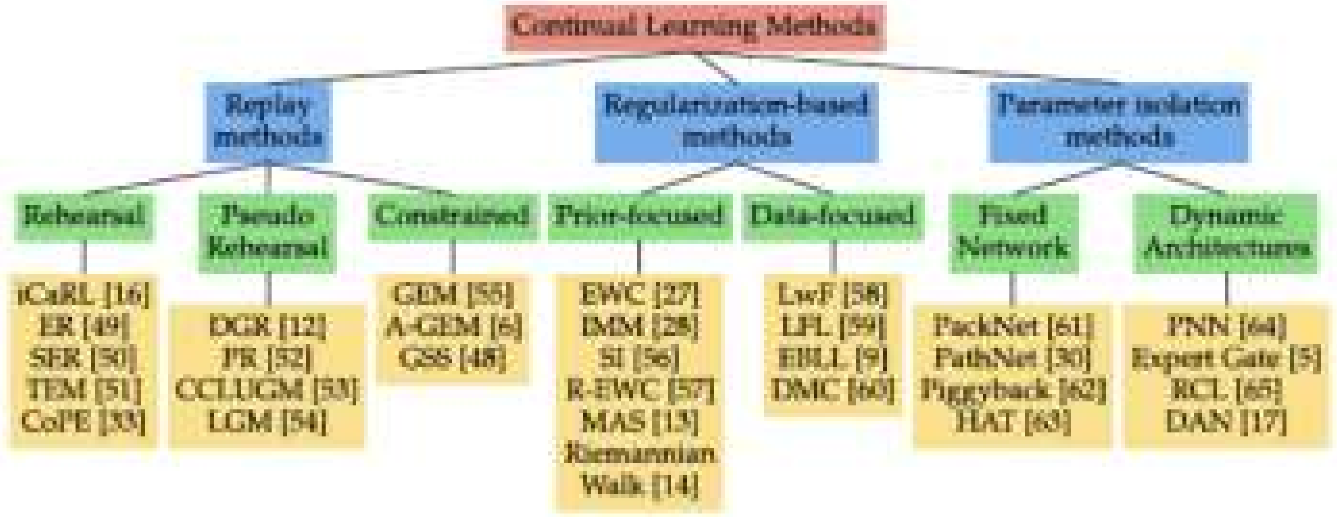


Category	Feature Extraction	Fine-Tuning	Duplicate and Fine-Tuning	Joint Training
New task performance	Medium	Good	Good	Good
Old task performance	Good	Bad	Good	Good
Training efficiency	Fast	Fast	Fast	Slow
Testing efficiency	Fast	Fast	Slow	Fast
Storage requirement	Medium	Medium	Large	Large
Require previous task data	No	No	No	Yes

Selected methods

Taxonomy☺





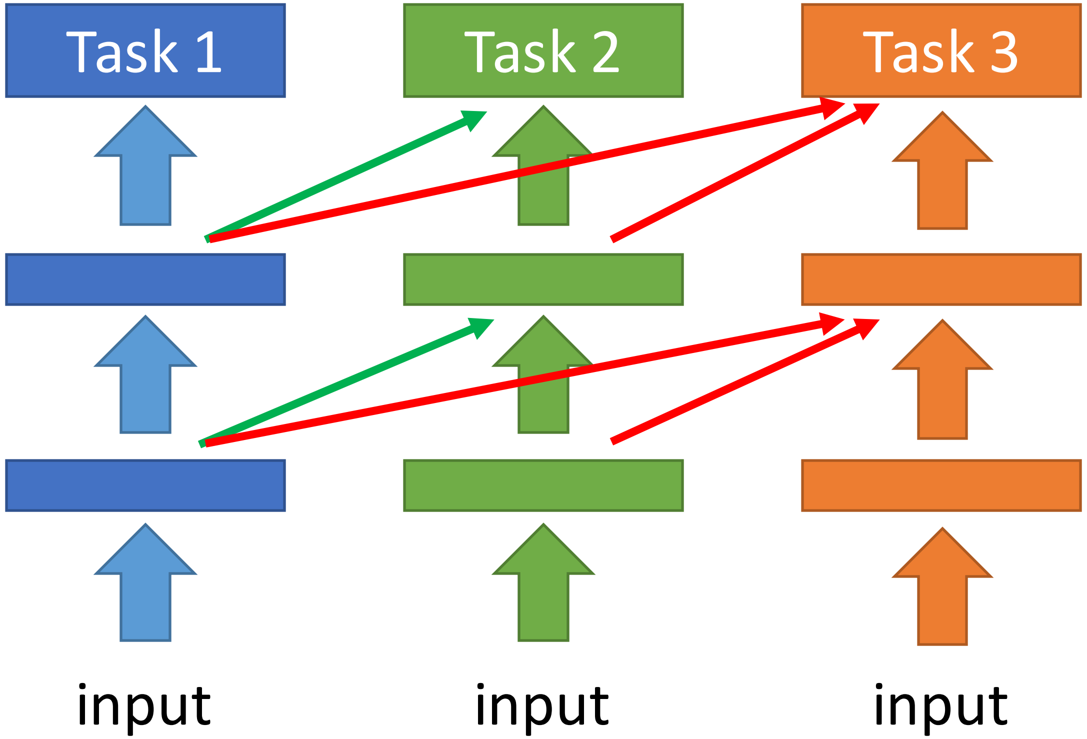
Lesort et al., Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges, Information Fusion, 2020

Matthias De Lange eta l., A Continual Learning Survey: Defying Forgetting in Classification Tasks, IEEE T PAMI, 2021

A straightforward way to do train model for lifelong ML is to explicitly define a different models per task.

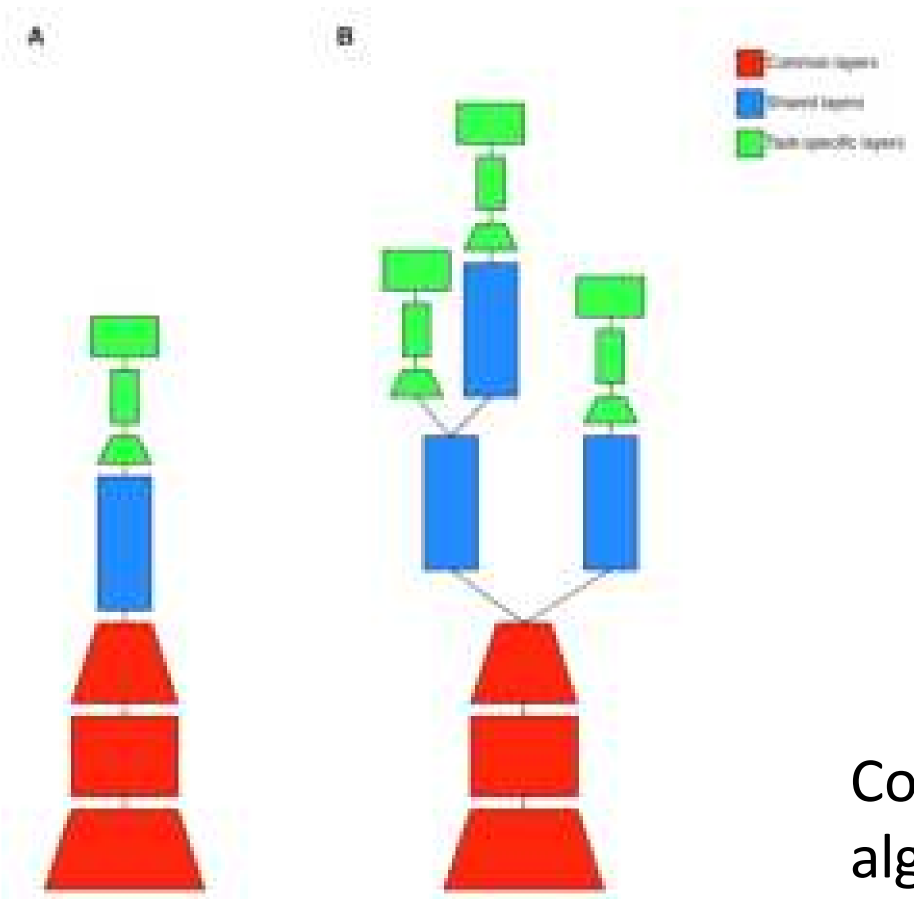
Progressive neural networks were proposed by Rusu et al. to explicitly tackle catastrophic forgetting for the problem of LL. The idea is to keep a pool of pretrained models as knowledge, and use lateral connections between them to adapt to the new task.

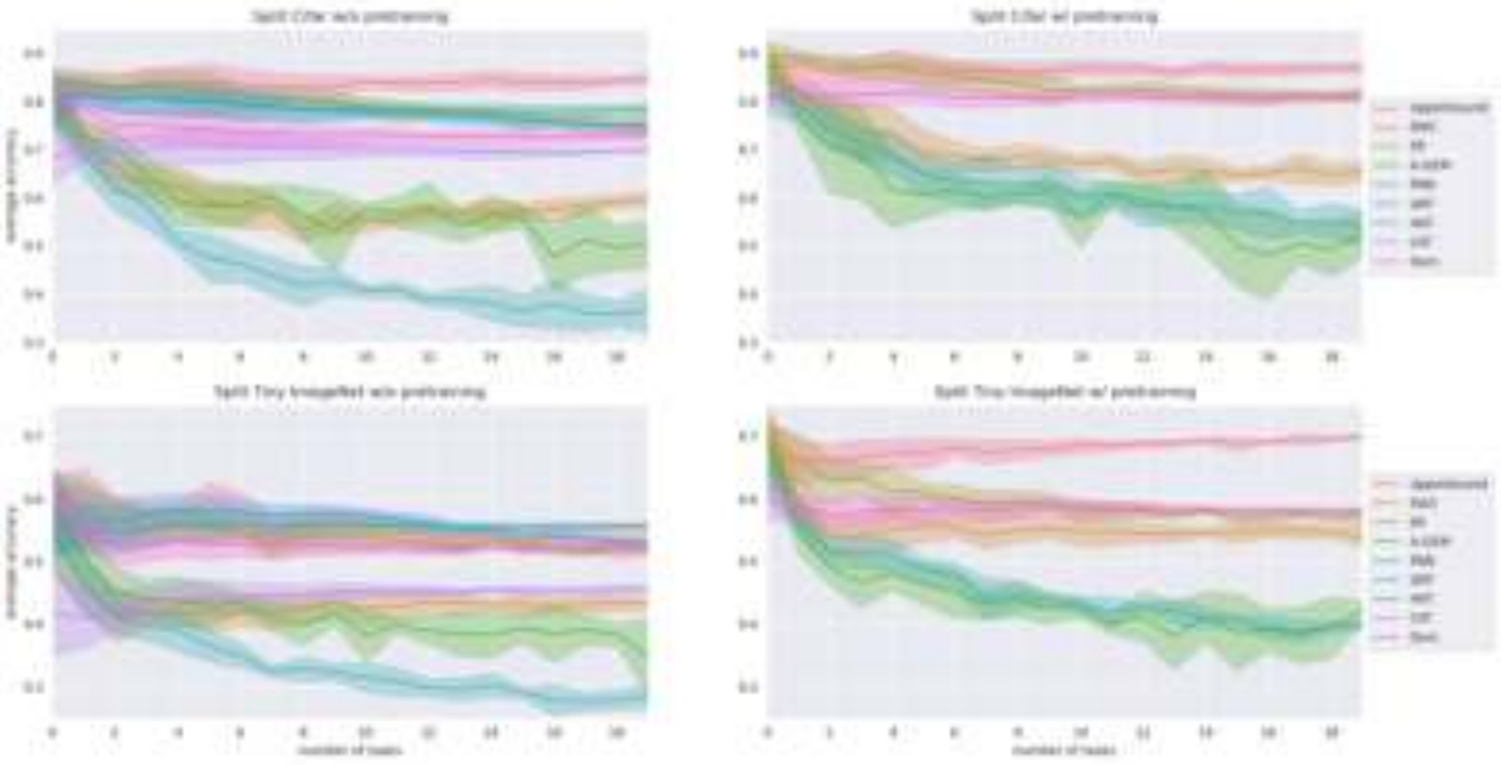
https://arxiv.org/abs/1606.04671



- We propose a novel approach based on adding new layers on top of existing ones to enable the forward transfer of knowledge and adapting previously learned representations.
- We employ a method of determining the most similar tasks for selecting the best location in our network to add new nodes with trainable parameters.
- This approach allows for creating a tree-like model, where each node is a set of neural network parameters dedicated to a specific task.

J. Kozal, M.Woźniak, Increasing Depth of Neural Networks for Life-long Learning, Information Fusion 2023





Conducted experiments suggest that the introduced algorithm obtained good results, especially for datasets with distribution close to real images. In a more challenging setup with a single computer vision dataset as a separate task, our method outperforms Experience Replay.

- Several recent papers use similar strategy, with different approaches for selecting the parts of the network for each task[1].
- *Context-dependent Gating* randomly assigns which nodes participate in each task[2].
- Other approaches use evolutionary algorithms or gradient descent to learn which units to employ for each task.

These methods could be used only in the task incremental learning scenario, as task identity is required to select the correct task-specific components.

1. Shipeng Yan et al., Dynamically Expandable Representation for Class Incremental Learning, CVPR 2021

2.Joan Serra et al., Overcoming Catastrophic Forgetting with Hard Attention to the Task, ICML 2018

Regularized Optimization is an alternative strategy.

It still preferentially train a different part of the network for each task, but to always use the entire network for execution.

It estimates (for all parameters of the network) how important they are for the previously learned tasks and penalize future changes to them accordingly (i.e., learning is slowed down for parts of the network important for previous tasks).

Kirkpatrick et al. proposed Elastic Weight Consolidation (EWC) https://arxiv.org/abs/1612.00796

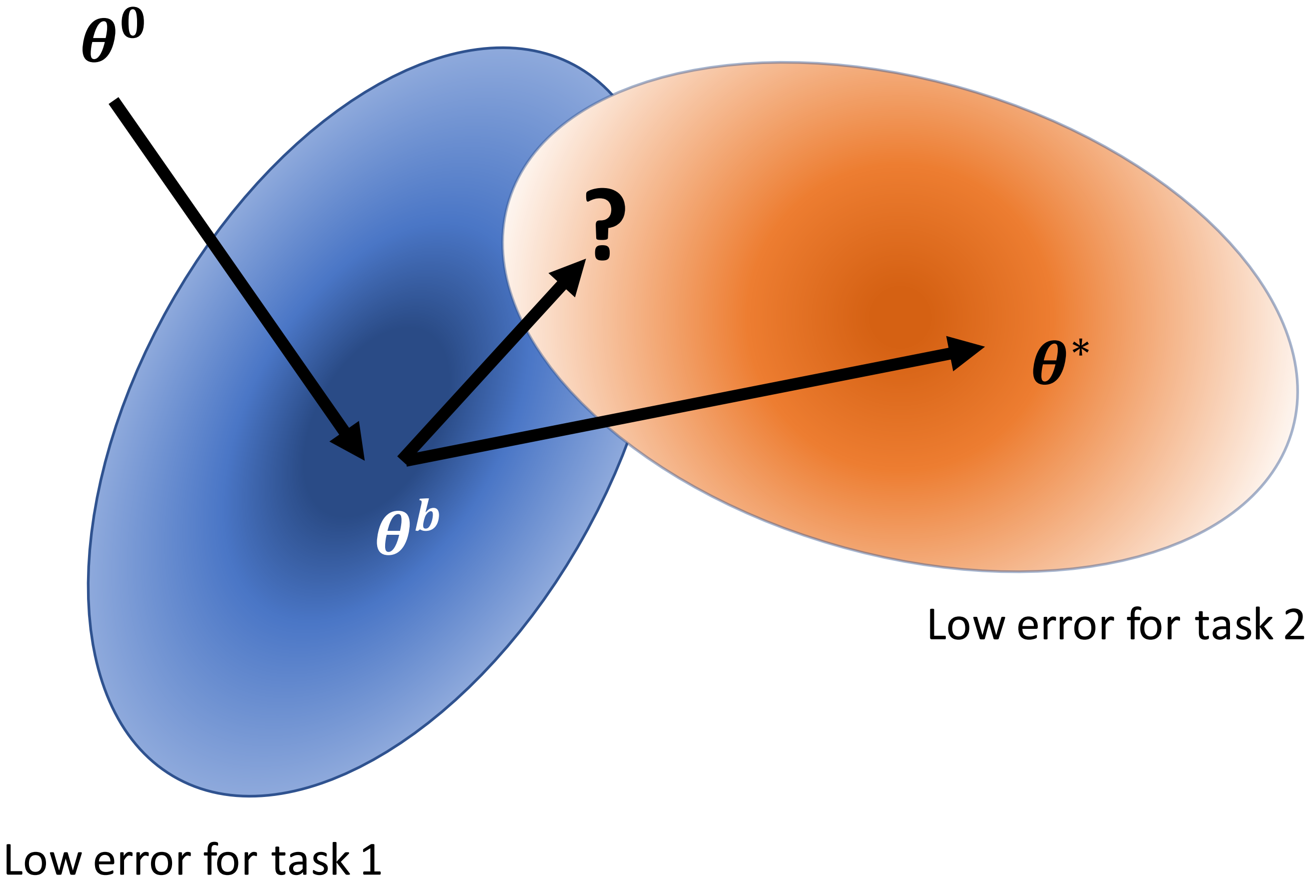
Gradient Episodic Memory (GEM) https://arxiv.org/abs/1706.08840

Synaptic Intelligence (SI) https://arxiv.org/abs/1703.04200

EWC was inspired by human brain in which synaptic consolidation enables continual learning by reducing the plasticity of synapses related to previous learned tasks. It mitigates catastrophic forgetting in neural networks.

Plasticity is the main cause of catastrophic forgetting since the weights learned in the previous tasks can be easily modified given a new task.

More precisely, plasticity of weights that are closely related to previous tasks is more prone to catastrophic forgetting than plasticity of weights that are loosely connected to previous tasks.



Some parameters are important to the previous tasks, then change the unimportant parameters.

$\boldsymbol{\theta^b}$ is the model learned from the previous tasks.

Each parameter θ_i^b has a “guard” b_i

Loss for current task

How important this parameter is

$$\underline{L'(\boldsymbol{\theta})} = \underline{L(\boldsymbol{\theta})} + \lambda \sum_i \underline{b_i}\left(\underline{\theta_i} - \underline{\theta_i^b}\right)^2$$

Loss to be optimized

Parameters to be learning

Parameters learned from previous task

Some parameters are important to the previous tasks, then change the unimportant parameters.

$\boldsymbol{\theta^b}$ is the model learned from the previous tasks.

Each parameter θ_i^b has a “guard” b_i

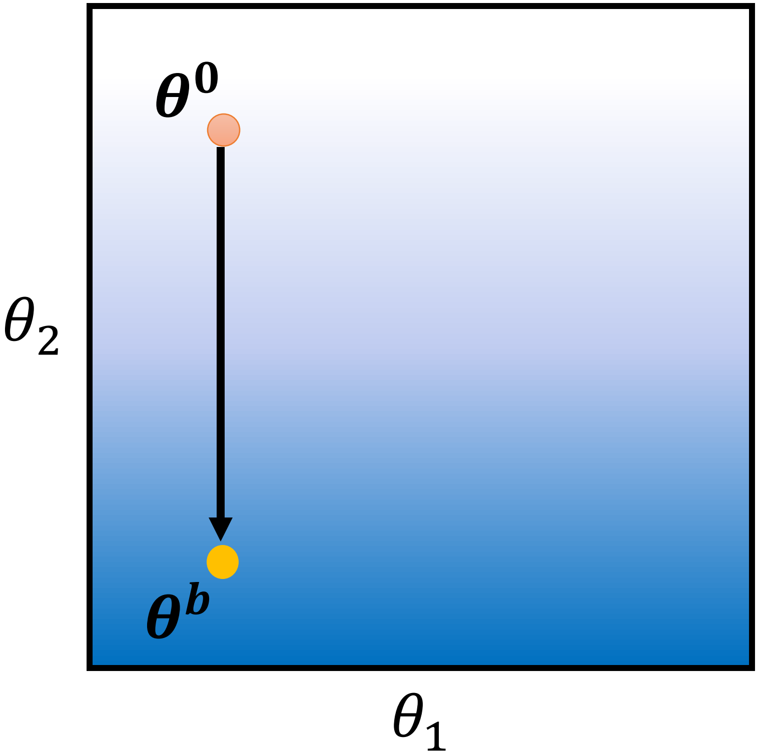
$\boldsymbol{\theta}$ should be close to $\boldsymbol{\theta^b}$ in certain directions.

$$L'(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \lambda \boxed{\sum_i b_i \left(\theta_i - \theta_i^b\right)^2}$$

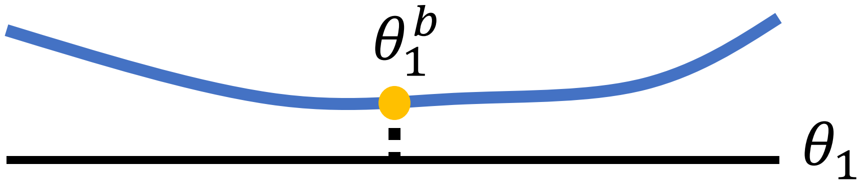
If $b_i = 0$, there is no constraint on θ_i → Catastrophic Forgetting

If $b_i = \infty$, θ_i would always be equal to θ_i^b → Intransigence

Task 1

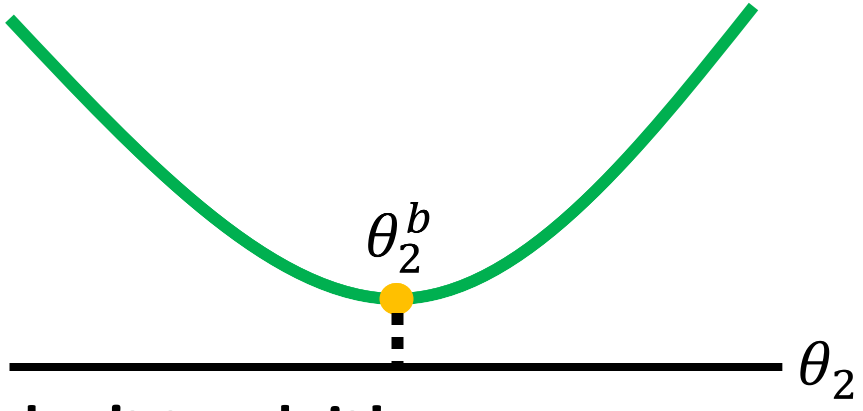


Each parameter has a “guard” b_i



can be changed ☺

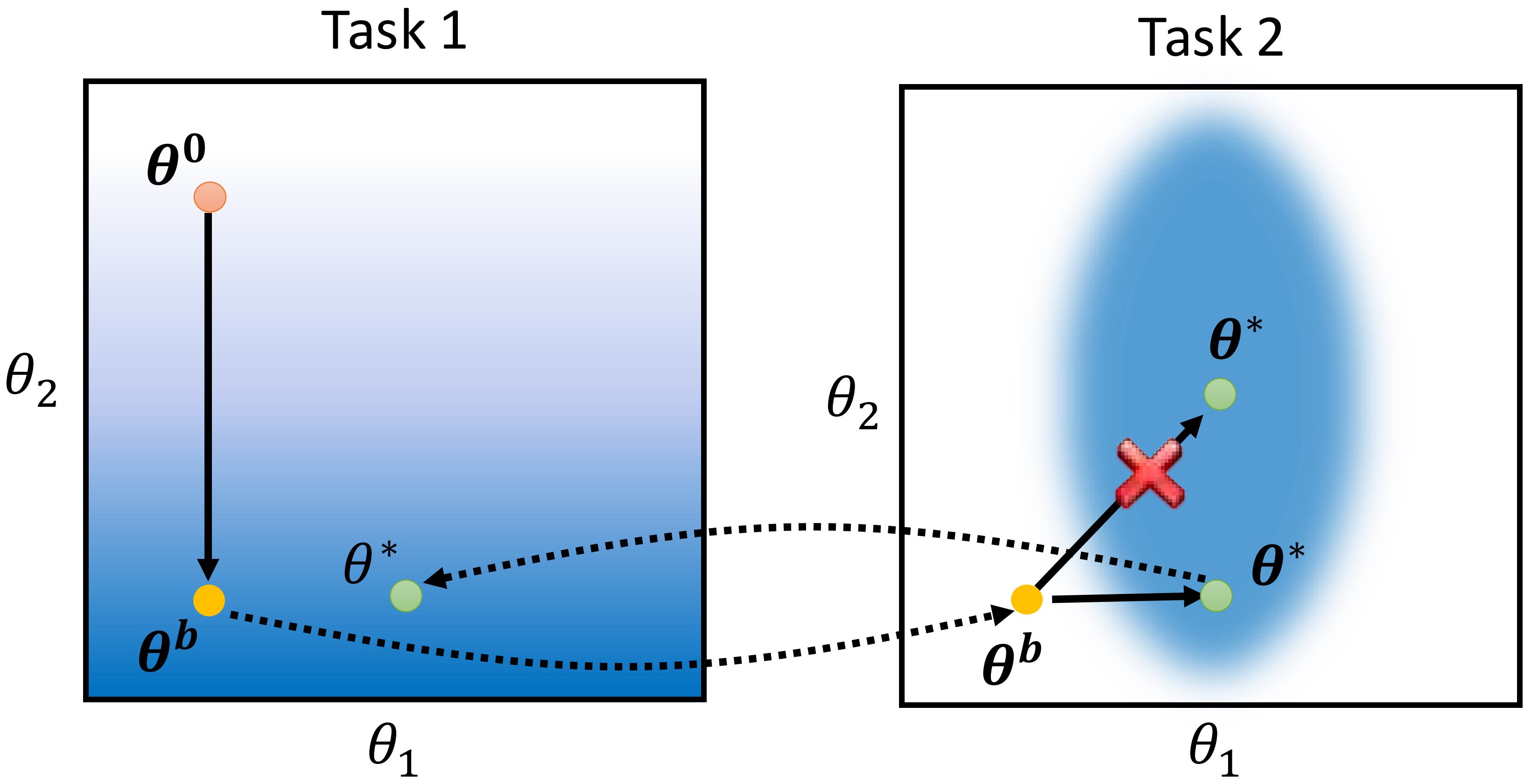
➡ b_1 is small



don’t touch it!

➡ b_2 is large

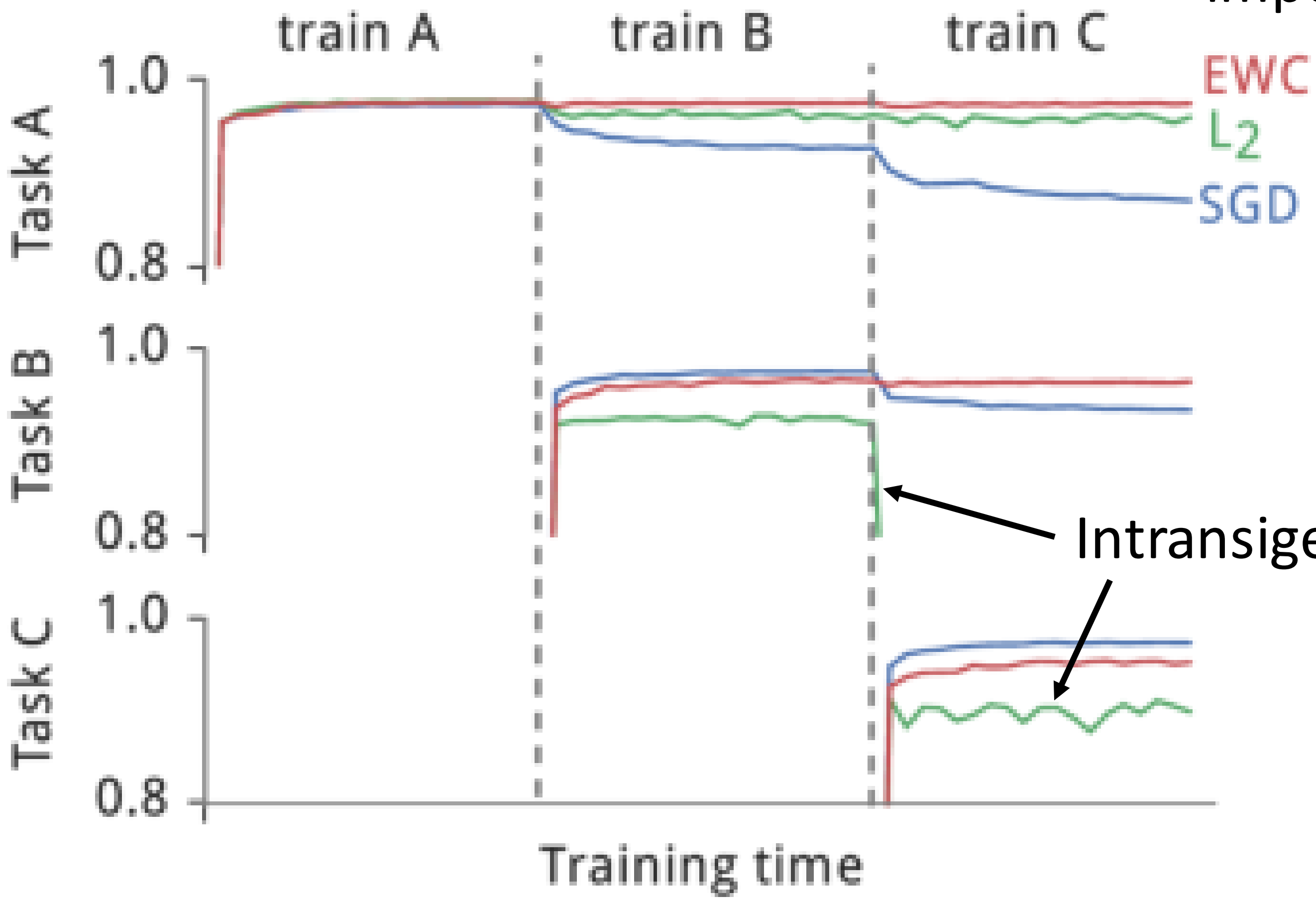
Example given by Hung-yi Lee, NTU



b_1 is small, while b_2 is large.

(We can modify θ_1, but do not change θ_2.)

b_i represents importance



$b_i = 1$

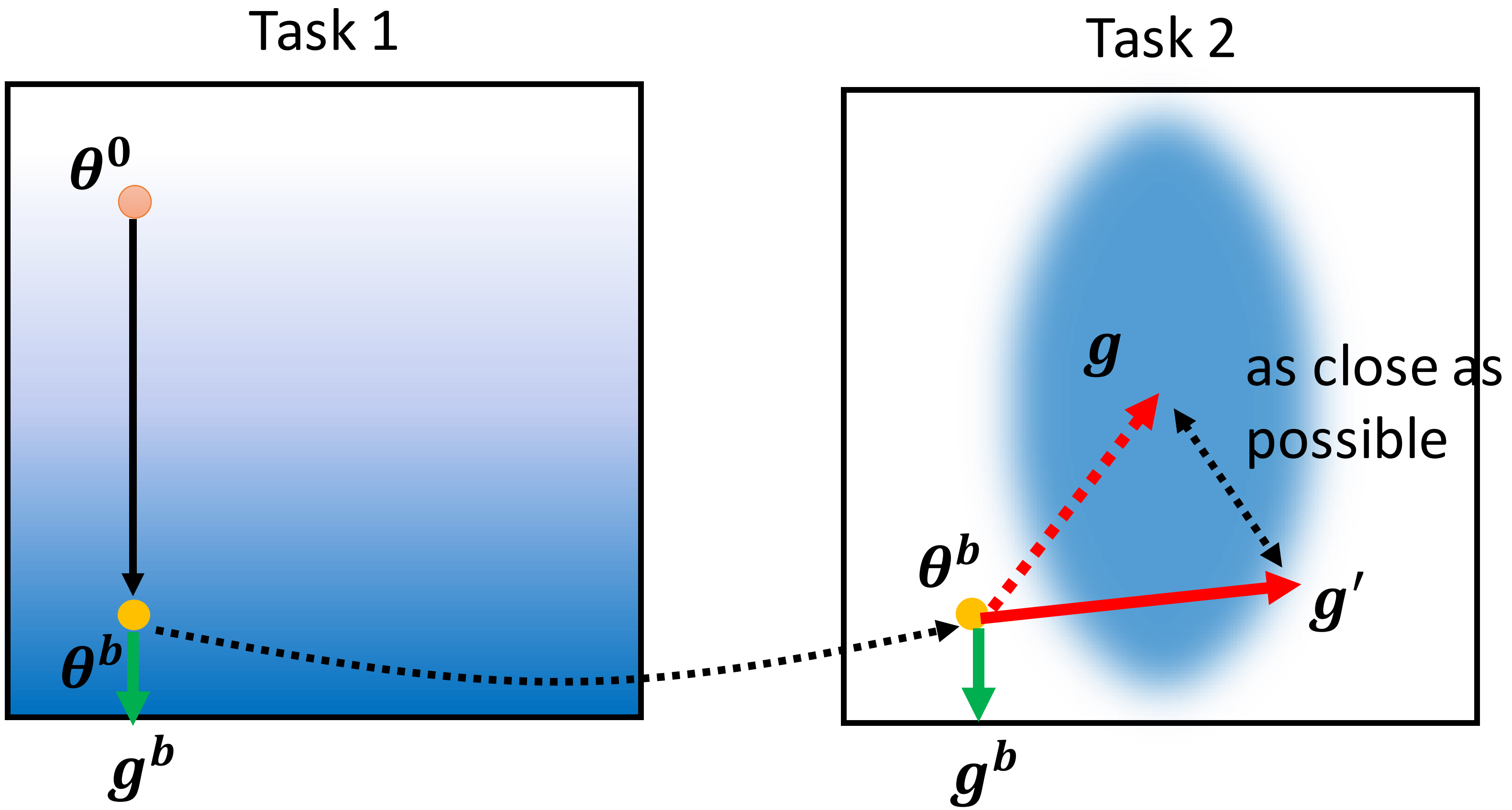
$b_i = 0$

Intransigence

MNIST permutation https://arxiv.org/abs/1612.00796

Gradient Episodic Memory (GEM)

https://arxiv.org/abs/1706.08840



$$g' \cdot g^b \geq 0$$

⇢ : negative gradient of current task

→ : negative gradient of previous task

→ : update direction

Need the data from the previous tasks

Learning without forgetting optimizes shared parameters and parameters for new task during new task learning with constrains that the predictions on the new task examples using old parameters and shared parameters do not shift too much.

Replay is an alternative strategy for combating catastrophic forgetting.

- It is to complement the training data for each new task to be learned with “pseudo-data” representative of the previous tasks.
- It takes the input data of the current task, label them using the model trained on the previous tasks, and use the resulting input-target pairs as pseudo-data.
- Disadvantage: possible memory constraints or privacy concerns.

Shin et al. proposed **Deep Generative Replay -** a continual learning method using replayed examples from a generative model without referring to the actual data of past tasks.

It is inspired by the suggestion that the hippocampus is better paired with a generative model than a replay buffer, because it is more than a simple experience replay buffer.

Shin H., Continual Learning with Deep Generative Replay
https://proceedings.neurips.cc/paper/2017/file/0efbe98067c6c73dba1250d2beaa81f9-Paper.pdf

- If it is possible to store data from previous tasks then we may use stored data as “exemplars” during execution.
- **iCaRL, incremental Classifier and Representation Learning** uses a neural network for feature extraction and performs classification based on a nearest-class-mean rule in that feature space, whereby the class means are calculated from the stored data.
- To protect the feature extractor network from becoming unsuitable for previously learned tasks, iCaRL also replays the stored data—as well as the current task inputs with a special form of distillation—during training of the feature extractor.

https://arxiv.org/abs/1611.07725

Korycki and Krawczyk proposed **Reactive Subspace Buffer** (RSB).

Majority of the experience replay methods do not control validity of stored examples.

Further improve adaptation by making memory buffers reactive to concept drifts.

Reactive Subspace Buffer:

- Tracking the current dominant classes in a given cluster.
- Switching labels between clusters.
- Splitting them.
- Sampling from the replay buffer based on purity.
- Retaining stable concepts.

Ł. Korycki, B. Krawczyk: *Class-Incremental Experience Replay for Continual Learning Under Concept Drift*. CVPR Workshops: 3649-3658 (2021)

Class incremental learning for contradictory attack detection

A model that processes this type of traffic often uses time windows taking consecutive packets, which are then encoded into an array.

To apply given perturbations to the sample, network packets are prepared to achieve a similar effect to applying noise to an image. Altering network packets to fool the system is a common practice.

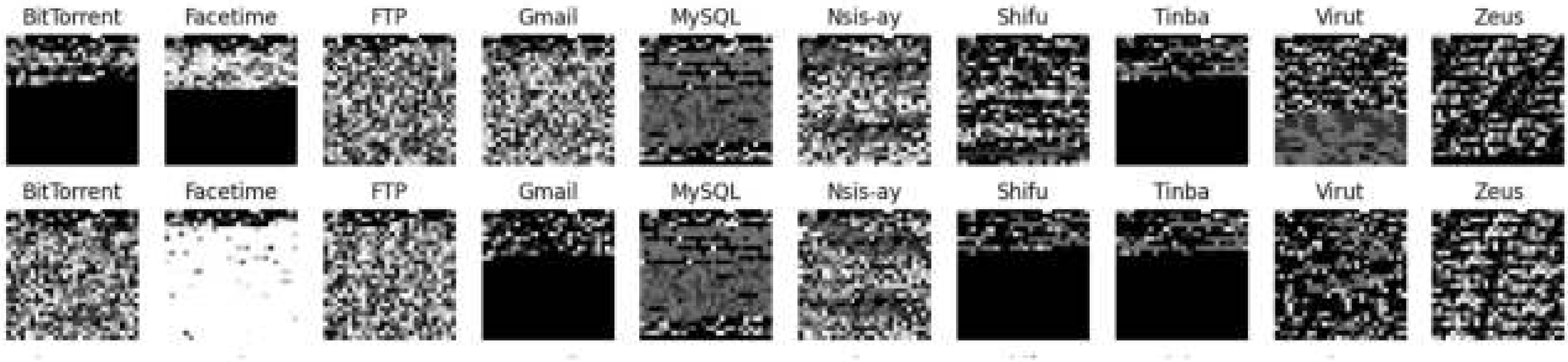
It involves altering the packets sent so that the security system (usually a firewall or other intrusion detection and prevention system) does not notice the suspicious traffic.

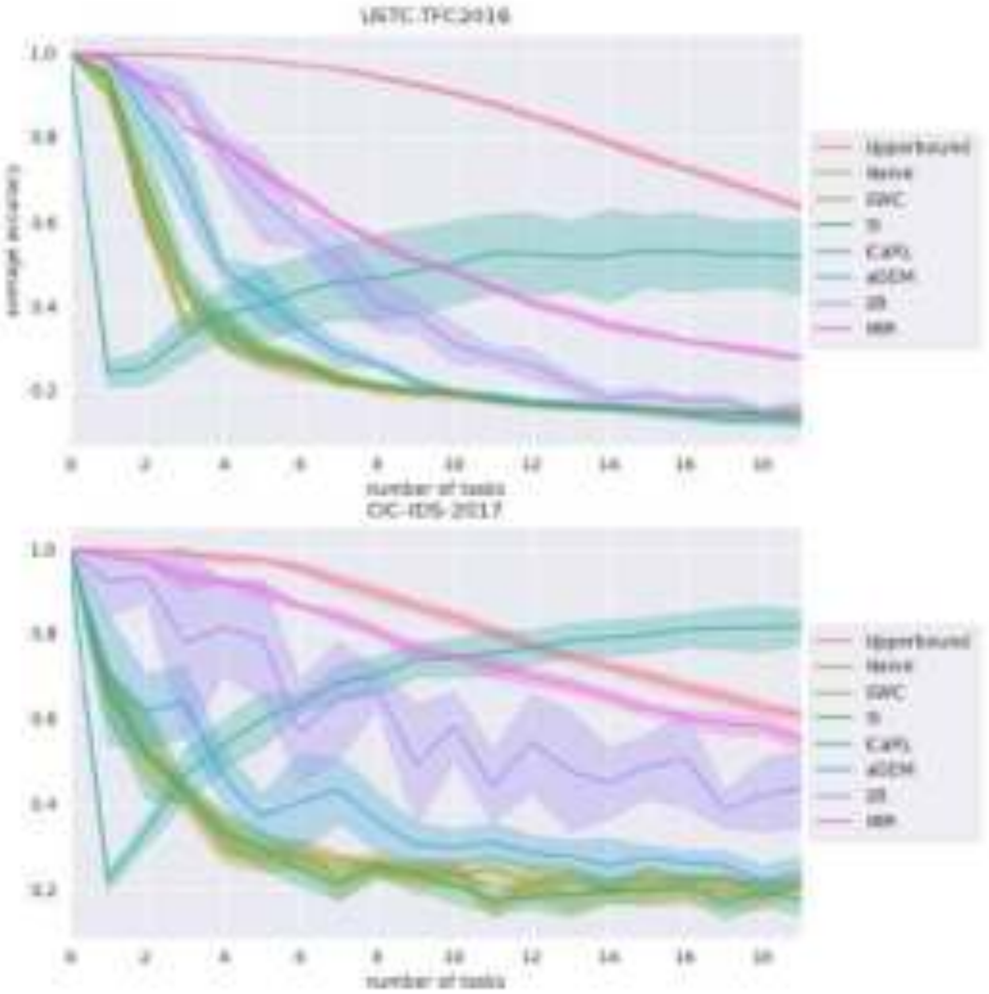
Changing packets most often requires the attacker to develop a new network packet or intercept an existing packet and edit the information in it

Kozal J. et al. Defending Network IDS against Adversarial Examples with Continual Learning IEEE ICDM 2023

Used generators: Hping, Ostinato,Scapy

Coding using USTC-TFC2016 Wei Wang et al. Malware traffic classification using convolutional neural network for representation learning (2017) https://ieeexplore.ieee.org/document/7899588





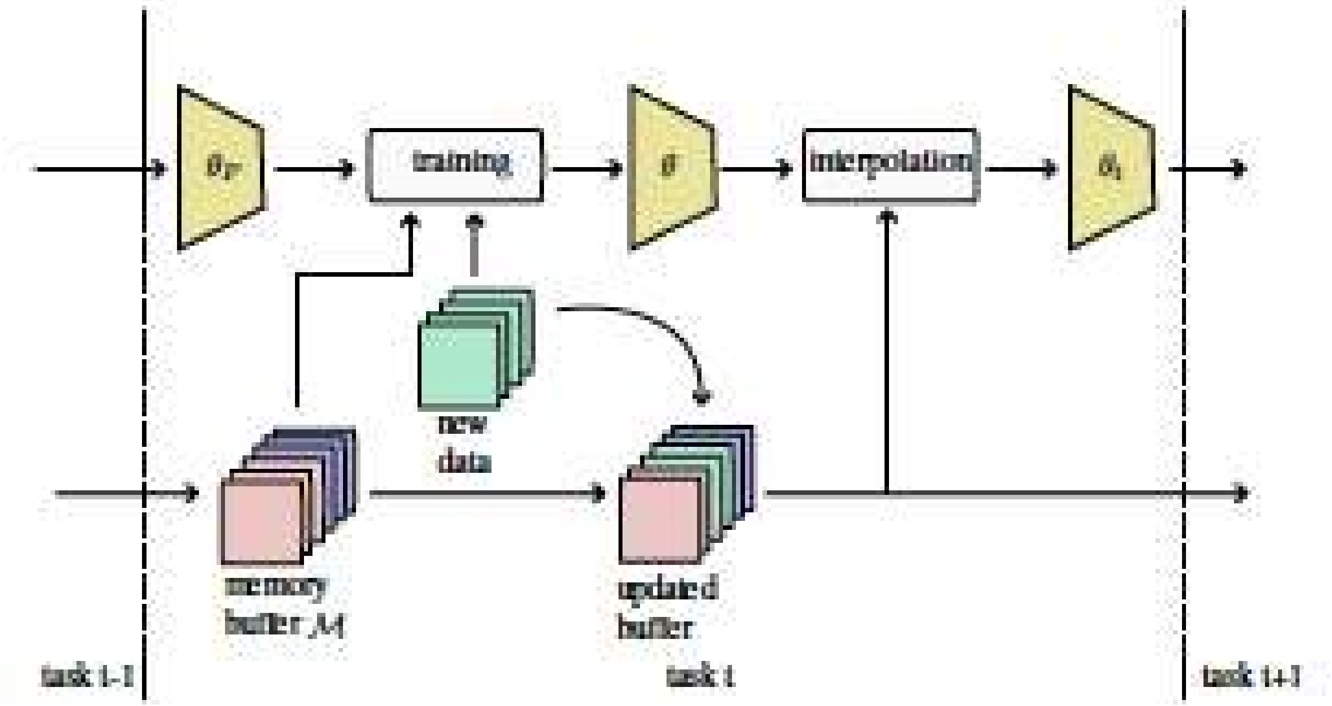
We confirmed that continual learning, in principle, could be used in the computer security domain to react to new attacks that appear over time and adapt existing models.

https://github.com/Rovlet/Adversarial-Incremental-Learning

We study the potential applications of recent findings from the field of weight interpolation in continual learning.

Based on recent weight interpolation techniques, we propose a remarkably simple continual learning algorithm that performs weight interpolation after each task to mitigate forgetting.

We base our approach on widely used experience replay methods.



Kozal J. et al. Continual Learning with Weight Interpolation, CVPR 2024 (submitted)

Challenges and conclusion

Evaluation of lifelong ML methods (metrics, protocols)

What we would like to measure?

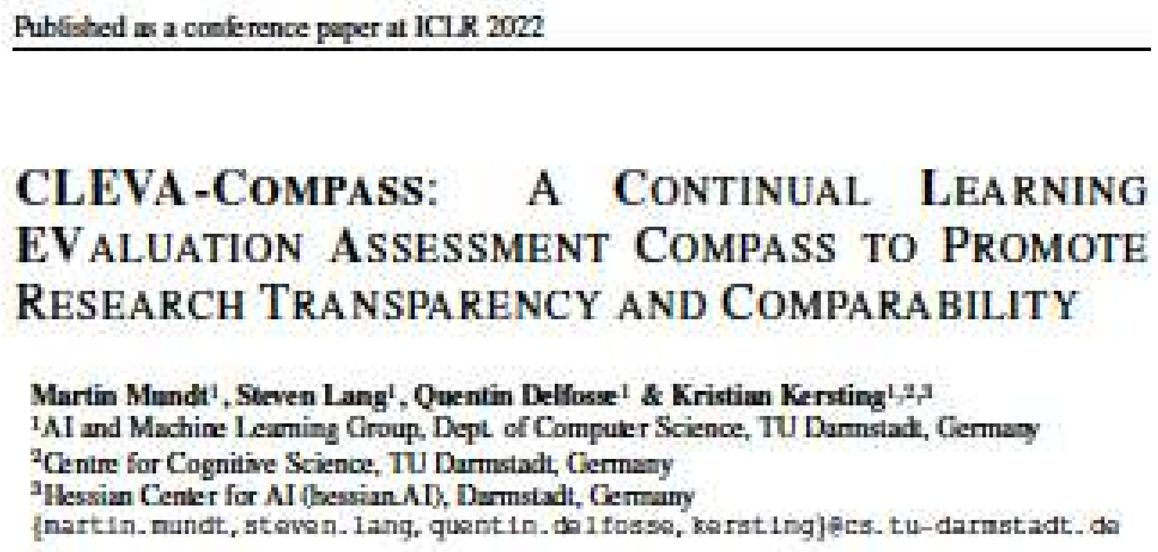
Performance on current experience
Performance on past experiences
Performance on future experiences
Resource consumption
Model size growth
Execution time
.....

What experimental protocol should be used?

Is the one protocol that fits all possible scenarios?

Lesort et al., Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges, Information Fusion, 2020

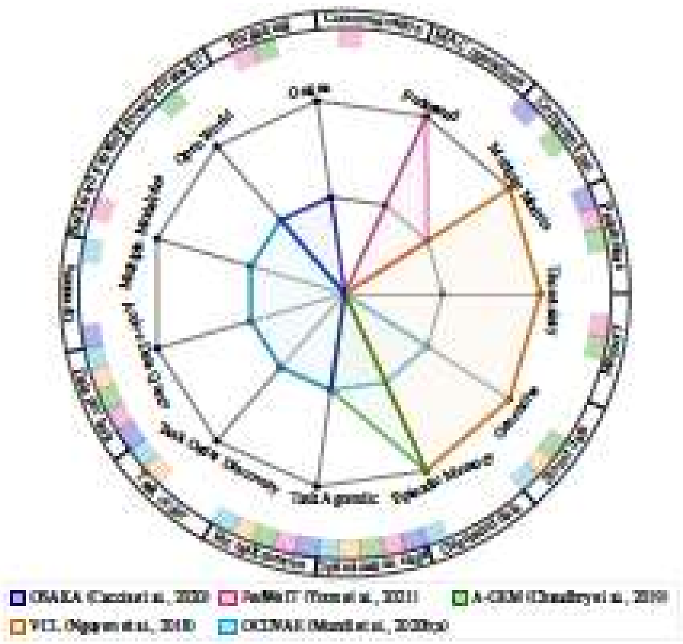
Evaluation of lifelong ML methods (metrics, protocols)



Published as a conference paper at ICLR 2022

CLEVA-COMPASS: A CONTINUAL LEARNING EVALUATION ASSESSMENT COMPASS TO PROMOTE RESEARCH TRANSPARENCY AND COMPARABILITY

Martin Mundt[1], Steven Lang[1], Quentin Delfosse[1] & Kristian Kersting[1,2,3]
[1]AI and Machine Learning Group, Dept. of Computer Science, TU Darmstadt, Germany
[2]Centre for Cognitive Science, TU Darmstadt, Germany
[3]Hessian Center for AI (hessian.AI), Darmstadt, Germany
{martin.mundt, steven.lang, quentin.delfosse, kersting}@cs.tu-darmstadt.de



It provides the visual means to identify how approaches are practically reported and how works can simultaneously be contextualized in the broader literature landscape. It provides an intuitive chart to understand the priorities of individual systems, where they resemble each other, and what elements are missing for a fair comparison.

New task definition – the current methods assume stationary model for each previously trained task

Lack of methods considered data updating or deleting

Taxonomy&formalism

CL is advanced conceptually, while there are quite few practical applications

What do we really expect and what can convince us to use CL?

Thrustworthy Continual Learning

- explainable,
- unbiased,
- reproducible,
- sustainable