Continual Learning: On Machines that can Learn Continually

Official Open-Access Course @ University of Pisa, ContinualAI, AIDA

Lecture 7: Methodologies [Part 3], Applications & Tools

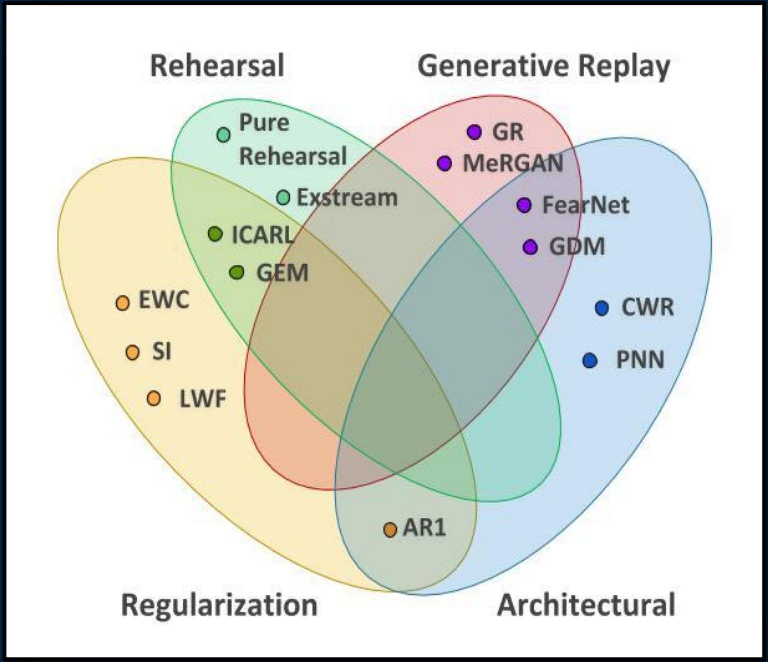
Vincenzo Lomonaco
University of Pisa & ContinualAI
vincenzo.lomonaco@unipi.it

TABLE OF CONTENTS

Hybrid Strategies

Why Hybrid?

- We explored several approaches for continual learning algorithms
- Each has **advantages** and **disadvantages** and works best within specific scenarios
- Such approaches are often **orthogonal with respect to each other**
- Biological learning systems seem to apply **several approaches** for learning continually
- **Hybrid approaches are underexplored** and may potentially find better Effectiveness-Efficiency trade-offs



Gradient Episodic Memory (GEM)

Key Aspects

- Can be seen as a **replay** and **regularization** method
- Each weight update is subject to an additional **inequality constraint** (loss on previous experiences can decrease but not increase)
- This allow for **positive backward transfer** and reduce overfitting
- Implemented as a **inner optimization** loop with **Quadratic Programming**

Algorithm 1 Training a GEM over an *ordered* continuum of data

```
procedure TRAIN(f_θ, Continuum_train, Continuum_test)
    M_t ← {} for all t = 1, ..., T.
    R ← 0 ∈ R^{T×T}.
    for t = 1, ..., T do:
        for (x, y) in Continuum_train(t) do
            M_t ← M_t ∪ (x, y)
            g ← ∇_θ ℓ(f_θ(x, t), y)
            g_k ← ∇_θ ℓ(f_θ, M_k) for all k < t
            g̃ ← PROJECT(g, g_1, ..., g_{t−1}), see (11).
            θ ← θ − αg̃.
        end for
        R_{t,:} ← EVALUATE(f_θ, Continuum_test)
    end for
    return f_θ, R
end procedure

procedure EVALUATE(f_θ, Continuum)
    r ← 0 ∈ R^T
    for k = 1, ..., T do
        r_k ← 0
        for (x, y) in Continuum(k) do
            r_k ← r_k + accuracy(f_θ(x, k), y)
        end for
        r_k ← r_k / len(Continuum(k))
    end for
    return r
end procedure
```

$$\ell(f_\theta, \mathcal{M}_k) = \frac{1}{|\mathcal{M}_k|} \sum_{(x_i,k,y_i)\in\mathcal{M}_k} \ell(f_\theta(x_i,k), y_i).$$

$$\begin{aligned} \text{minimize}_\theta \quad & \ell(f_\theta(x,t), y) \\ \text{subject to} \quad & \ell(f_\theta, \mathcal{M}_k) \le \ell(f_\theta^{t-1}, \mathcal{M}_k) \text{ for all } k < t, \end{aligned}$$

Gradient Episodic Memory for Continual Learning, Lopez-Paz et al, NeurIPS 2017.

Incremental Classifier and Representation Learning (iCaRL)

Key Aspects

- **Replay** and **Regularization** method
- **Distillation** for the **regularization of representation** learning (feature extractor)
- **Template matching** with **nearest prototype** (as a classifier)
- More **sophisticated examples management** through **herding**.
- **Difficult to scale**, inefficient example management with large memory sizes.

Algorithm 1 iCaRL CLASSIFY

input x // image to be classified
require $\mathcal{P} = (P_1, \dots, P_t)$ // class exemplar sets
require $\varphi : \mathcal{X} \to \mathbb{R}^d$ // feature map
  for $y = 1, \dots, t$ **do**
    $\mu_y \leftarrow \frac{1}{|P_y|} \sum_{p \in P_y} \varphi(p)$ // mean-of-exemplars
  end for
  $y^* \leftarrow \operatorname*{argmin}_{y=1,\dots,t} \|\varphi(x) - \mu_y\|$ // nearest prototype
output class label y^*

Algorithm 2 iCaRL INCREMENTALTRAIN

input $X^s, \dots, X^t$ // training examples in per-class sets
input K // memory size
require Θ // current model parameters
require $\mathcal{P} = (P_1, \dots, P_{s-1})$ // current exemplar sets
  $\Theta \leftarrow \text{UPDATEREPRESENTATION}(X^s, \dots, X^t; \mathcal{P}, \Theta)$
  $m \leftarrow K/t$ // number of exemplars per class
  for $y = 1, \dots, s-1$ **do**
    $P_y \leftarrow \text{REDUCEEXEMPLARSET}(P_y, m)$
  end for
  for $y = s, \dots, t$ **do**
    $P_y \leftarrow \text{CONSTRUCTEXEMPLARSET}(X_y, m, \Theta)$
  end for
  $\mathcal{P} \leftarrow (P_1, \dots, P_t)$ // new exemplar sets

iCaRL: Incremental Classifier and Representation Learning, Rebuffi et al, 2017.

Progress & Compress (P&C)

Key Aspects

- **Architectural** and **Regularization** strategy
- **Distillation**, **Elastic Weight Consolidation** regularization and **lateral connections** (similar to Progressive Neural Networks)
- Alternation of **progress** (P) and **compress** (C) phases, for high plasticity and consolidation
- This paper introduces also an **Online-EWC implementation** description.

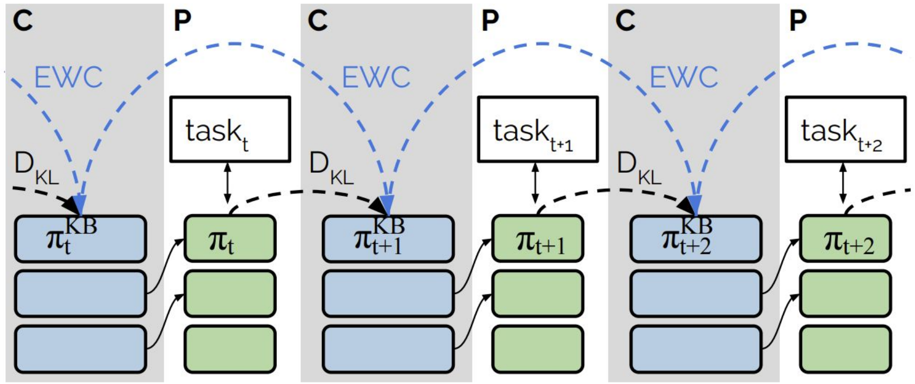


Figure 1. Illustration of the Progress & Compress learning process. In the compress phases (C), the policy learnt most recently by the active column (green) is distilled to the knowledge base (blue) while protecting previous contents with EWC (Elastic Weight Consolidation). In the progress phases (P), new tasks are learnt by the active column while reusing features from the knowledge base via lateral, layerwise connections.

Progress & Compress: A scalable framework for continual learning, Schwarz et al, 2018.

AR1: a Flexible Hybrid Strategy for Continual Learning

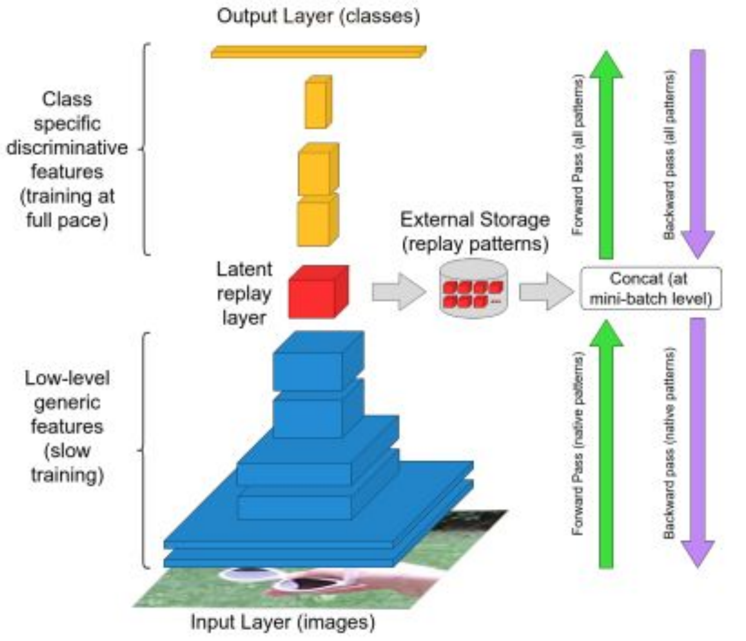


Figure 1: Architectural diagram of Latent Replay.

Key Ideas

- ***Architectural***, ***Regularization*** *and* ***Replay*** *components:*
  - *CWR* for the output layer (arch)*
  - *Online Synaptic Intelligence (reg)*
  - *Latent Replay (replay)*

$$\tilde{L}_\mu = L_\mu + \lambda \sum_k \Omega_k^\mu \,(\bar{\theta}_k - \theta_k)^2$$

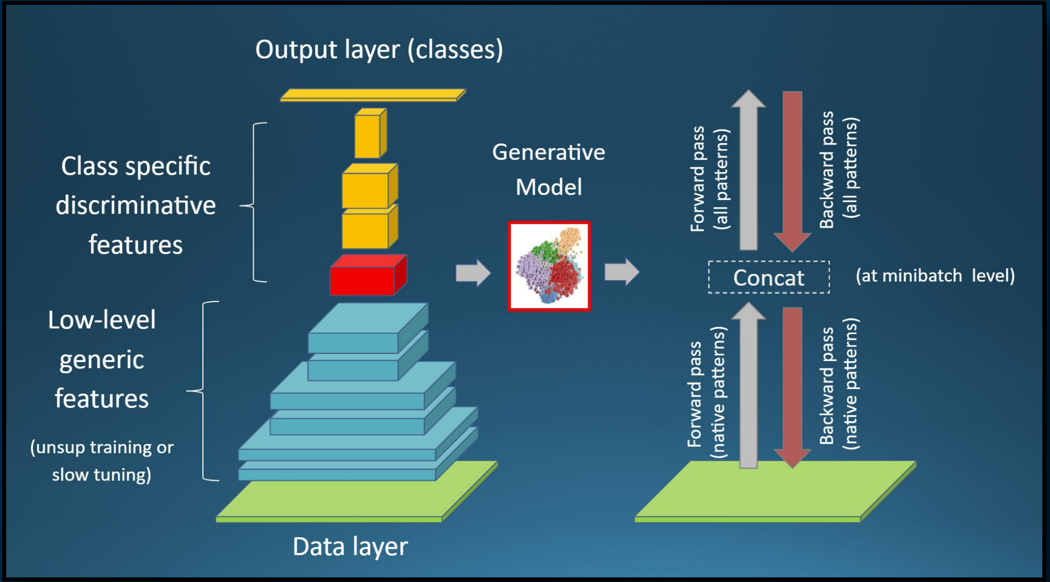
$$w_k^v = \int_{t^{\mu-1}}^{t^\mu} \frac{\partial L}{\partial \theta_k} \cdot \frac{\partial \theta_k}{\partial t}$$

D. Maltoni et al. ***Continuous Learning in Single-Incremental-Task Scenarios***, Neural Networks, 2019.
L. Pellegrini et al. ***Latent Replay for Real-Time Continual Learning***, IROS 2020.
V. Lomonaco et al. ***Rehearsal-Free Continual Learning over small I.I.D Batches***. CLVision at CVPR 2020.

AR1 and (Negative) Generative Replay?

Key Aspects

- **Generative Replay** is often difficult to scale (quality and diversity), what about **generative latent replay**?
- **Sharing weights** between the **discriminator** and the **generator** is possible
- **Incremental training of the generator** in the loop
- **Negative replay**: use generated patterns as negative examples only



Generative Negative Replay for Continual Learning, Graffiti et al, 2021.

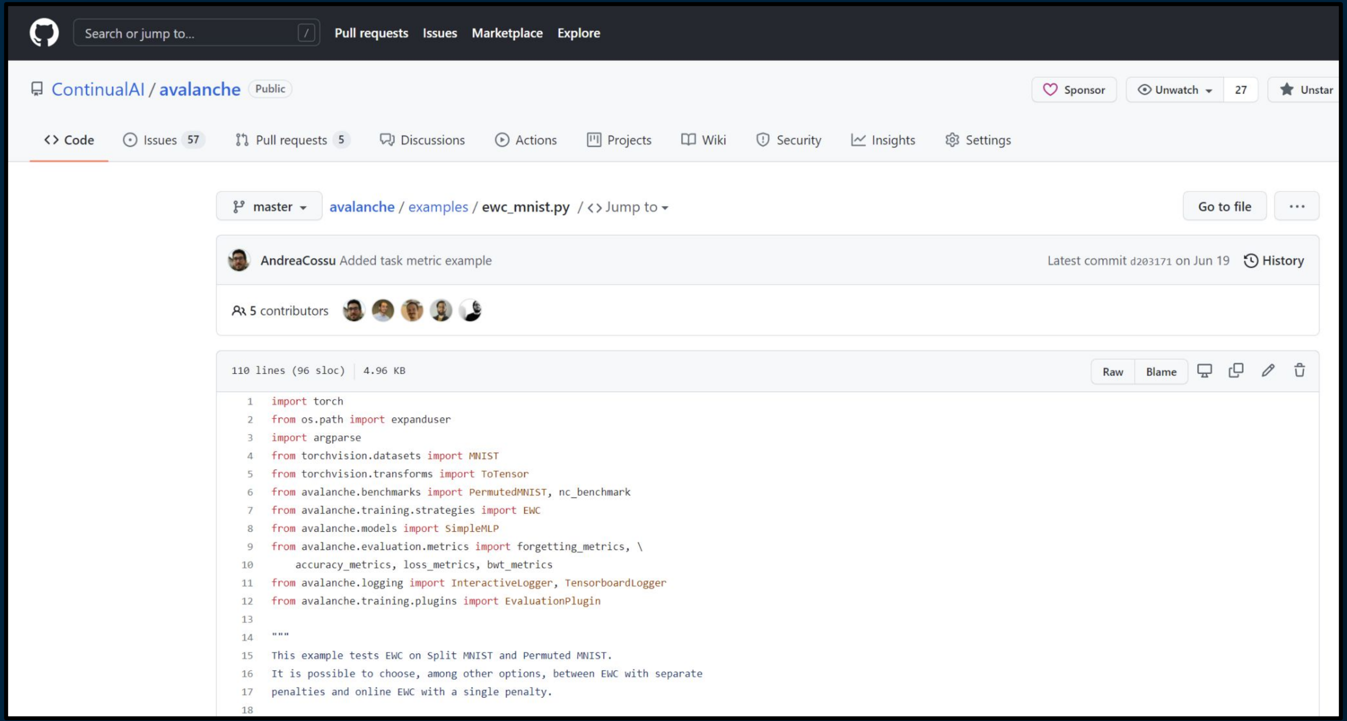
Summary & Next Steps

- **Hybrid approaches** are **more complex and more difficult to parametrize** in general but they can provide **improved Effectiveness-Efficiency** trade-offs.
- Such approaches are still **not well investigated** but offer a nice path for future research explorations.
- They are often among the **winning approaches in continual learning challenges**.
- More **flexible and tunable algorithms** (and possible self-adjusting hybrid approaches) may be quite interesting to investigate.

CVPR 2020 Continual Learning in Computer Vision Competition: Approaches, Results, Current Challenges and Future Directions, Lomonaco et al, 2020.

Avalanche GEM, iCaRL, AR1 Implementation

Hands-on Session!



https://github.com/ContinualAI/avalanche

Continual Learning

Applications

Continual Learning Applications

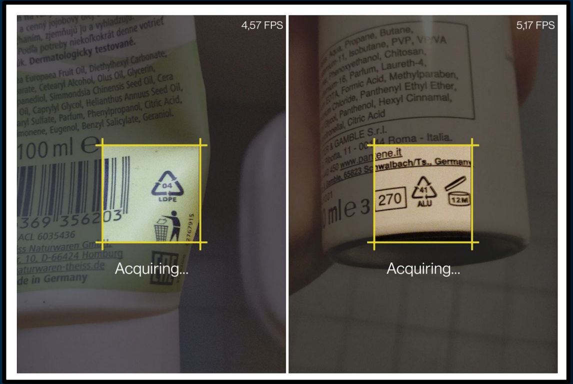
Main Possibilities (Grouped by "Where" Computation Happen)

- **Edge**
    - **Embedded systems and Robotics:** +privacy, +efficiency, +fast adaptation, +on the edge, -Internet connection (e.g. Autonomous Cars, Robotics Arms/Hands)
- **Cloud**
    - **AutoML and CI systems for AI models:** +scalability, +efficiency, +fast adaptation, -energy consumption, -$$$ (e.g. Recommendation Systems)
- **Continuum Edge-Cloud**
    - **Pervasive AI systems**: Efficient Communication, fluid & dynamic computation
    - **Neural Patches**: +security patches, +fairness patches, +fast update
    - **Continual Distributed Learning**: understudied relationship with parallel and federated learning

On-Device Personalization without Forgetting

L. Pellegrini et al. ***Latent Replay for Real-Time Continual Learning***, IROS 2020.
L. Pellegrini et al. ***Continual Learning at the Edge: Real-Time Training on Smartphone Devices***. ESANN, 2021.
G. Demosthenous et al. ***Continual Learning on the Edge with TensorFlow Lite***. *arXiv 2021.*
L. Ravaglia et al. ***Memory-Latency-Accuracy Trade-offs for Continual Learning on a RISC-V Extreme-Edge Node***. *SiPS 2020.*

Use-Case: Recycling Codes Recognition



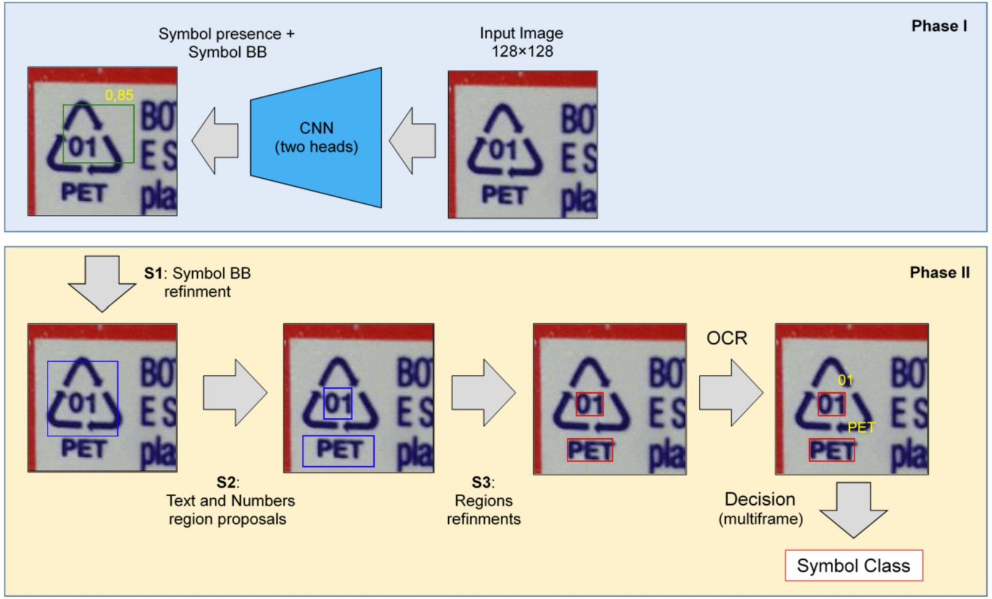


Figure 4. Overall schema of the proposed approach.

A Weakly Supervised Approach for Recycling Code Recognition, Pellegrini et al, 2021.

Continual Learning in Production

- **Guarantee QoS** and **reduce uncontrolled outcomes**
- **Performance is the reference KPI** for many application
- **Training memory and computation overheads are often not a concern**, real-time inference is.
- **On-the-fly personalized learning** vs **Adaptation** does **not** necessarily **involve training**
- Interesting to understand how CL would fit into more articulated **MLOps pipeline**

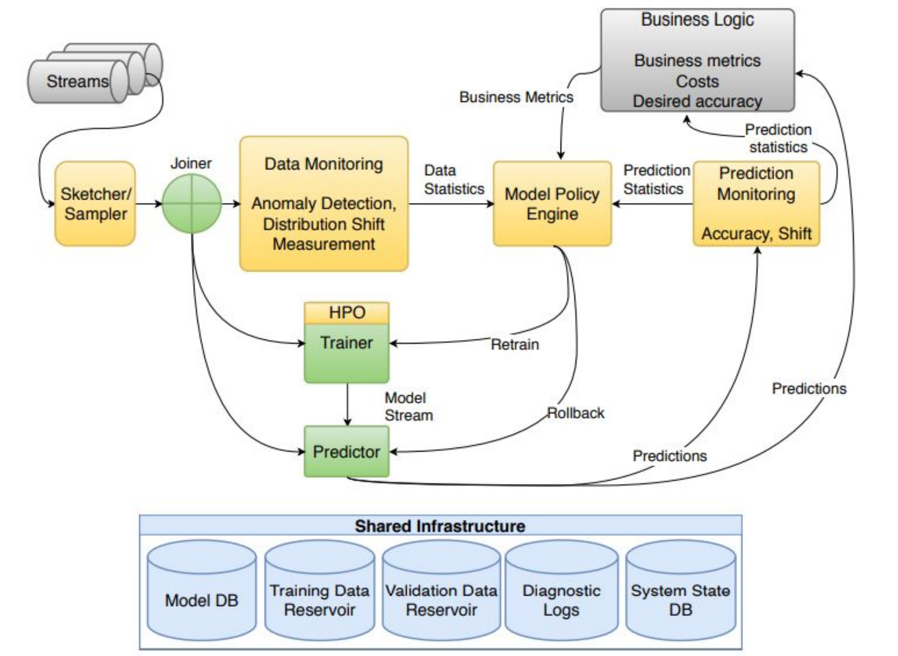
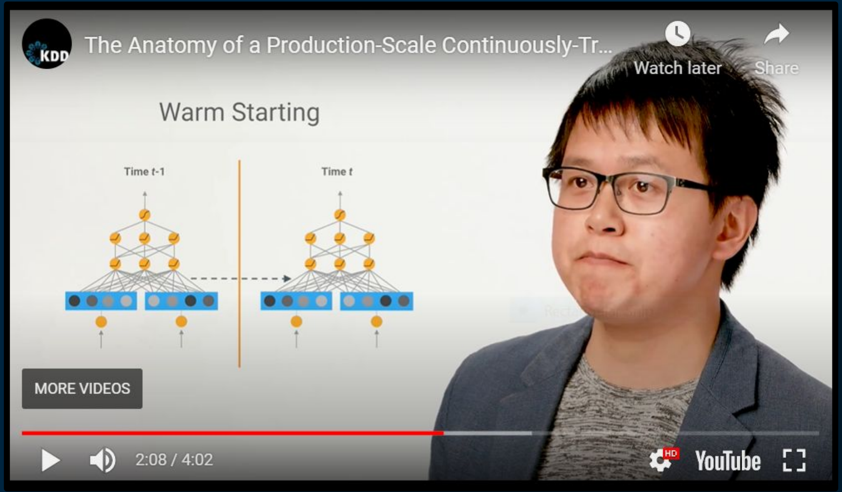


Figure 1: Data flow in the Auto-Adaptive Machine Learning architecture. See text for details.

T. Diethe et al. ***Continual Learning in Practise***. Continual Learning Workshop at NeurIPS 2018.

Use-Cases: Google Play

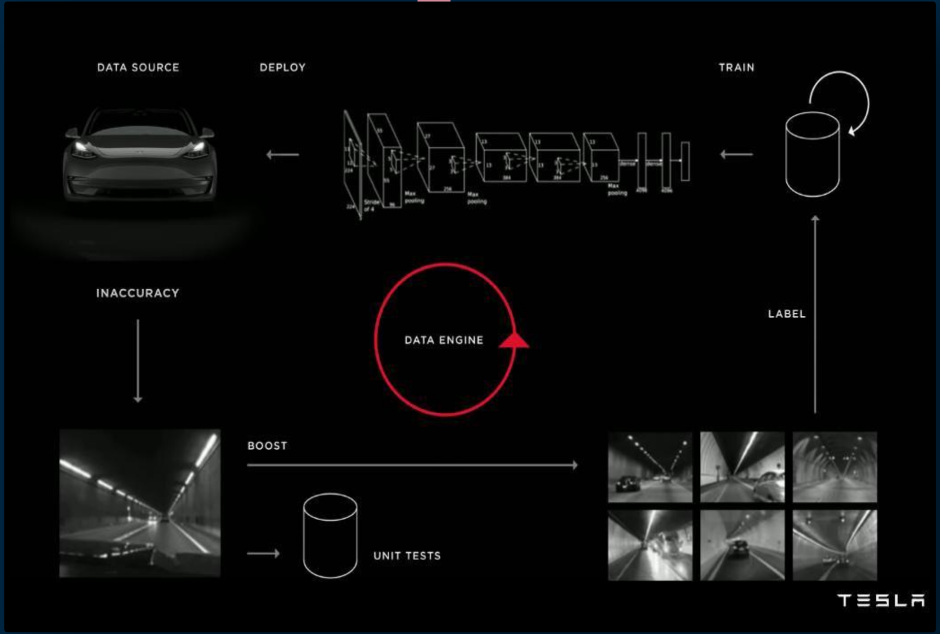


- **A continual learning system** based on **Tensorflow Extended (TFX)**
- **Warm Starting** as a **simple continual learning strategy**
- Used for **Google Play recommendation engine**
- **Warm init** (starting from the latest trained model using all the data) is known to be **underperforming** w.r.t. the **Cumulative strategy**

V. Lomonaco. ***Continual Learning for Production Systems: The new "Agile" in the Machine Learning Era***. ContinualAI Publication, 2019.
D. Baylor et al. ***TFX: A TensorFlow-Based Production-Scale Machine Learning Platform***. KDD, 2017.

Use-Cases: Tesla

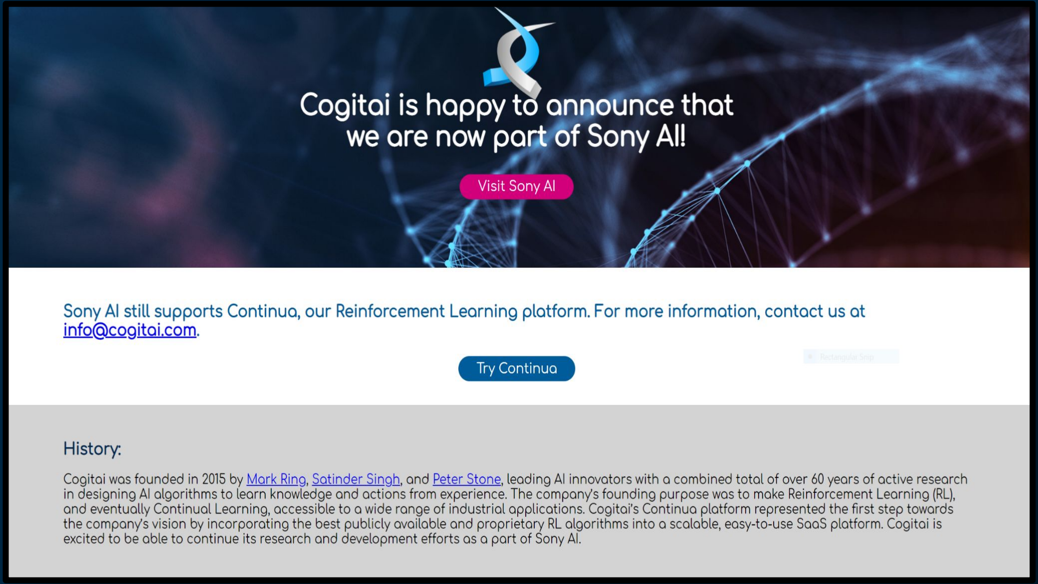
- A clear example of a **Continuous Training** and **Integration system**
- **Complex "Hydra" multi-head neural nets** for handling dozen of predictive tasks
- **Dual system networks at the edge**: one in use one for testing
- **Focused data collection strategy** for *"Curriculum Learning"*
- Not clear if any continual learning strategy is used at all...



V. Lomonaco. ***Continual Learning for Production Systems: The new "Agile" in the Machine Learning Era***. ContinualAI Publication, 2019.
A.Karpathy. ***Building the Software 2.0 Stack***. Spark+AI Summit, 2018.
Tesla's HydraNets - How Tesla's Autopilot Works

Startups Example: Cogitai

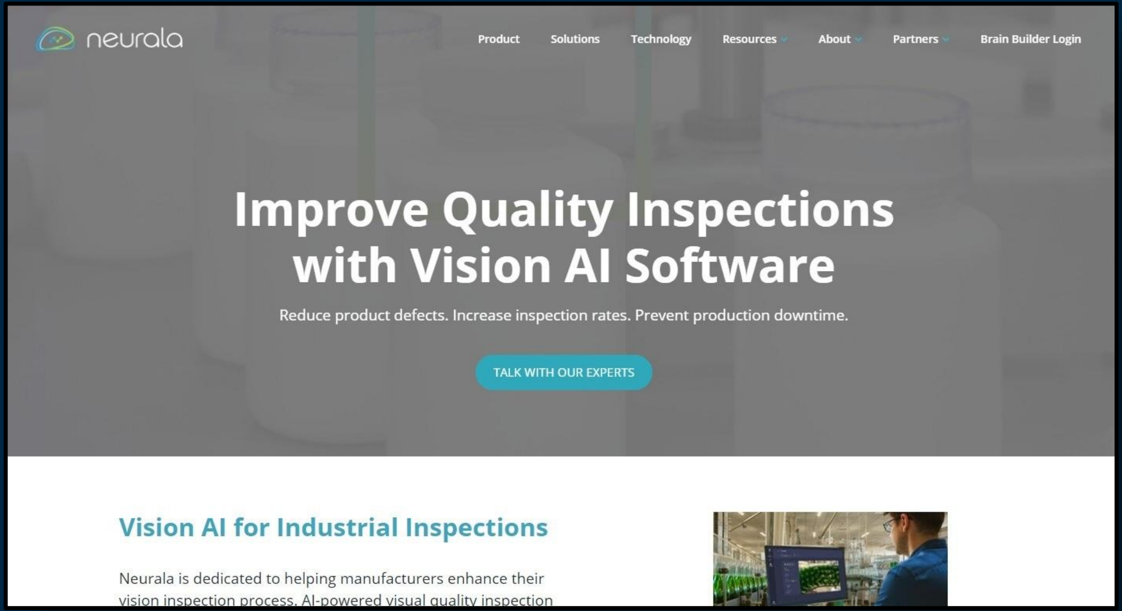
- The "*only Continual Learning startup on earth*" by **Mark Ring**, based on Ring Continual Learning specific formulation (different from *Lifelong Learning*)
- **Founded in 2015, sold in 2016 to Sony AI**
- **CLaaS platform**, mostly focused on Reinforcement Learning algorithms
- Not clear if the platform "**continua**" has being discontinued…



Continual Learning Workshop @ NeurIPS 2018 - Introduction by Mark Ring
https://www.cogitai.com

Startups Example: Neurala

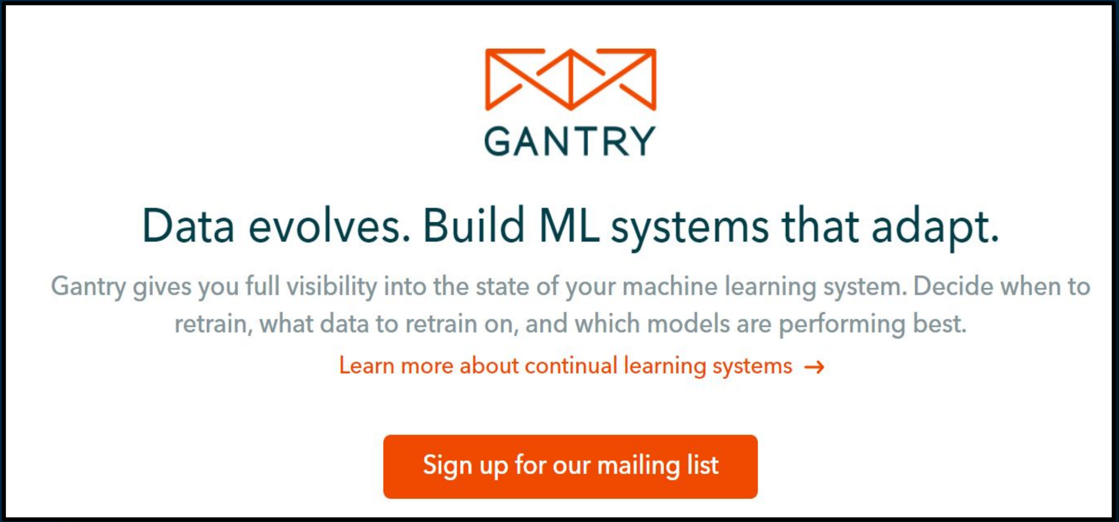
- Founded **in 2013** by **Max Versace** and team in Boston
- Attracted more than **25M$ in funding**
- Focus on **Lifelong-Learning custom technology and patent**
- Now focusing on **Quality Inspections Vision Application**



https://www.neurala.com
Neurala Lifelong DNN tech

Startups Example: Gantry

- Sort of **OpenAI spin-off** company
- Focusing on **MLOps infrastructure** for evolving data
- **Still in the early days**, but interesting set of investors and key human resources
- It suggests large opening in this **fast emerging market**

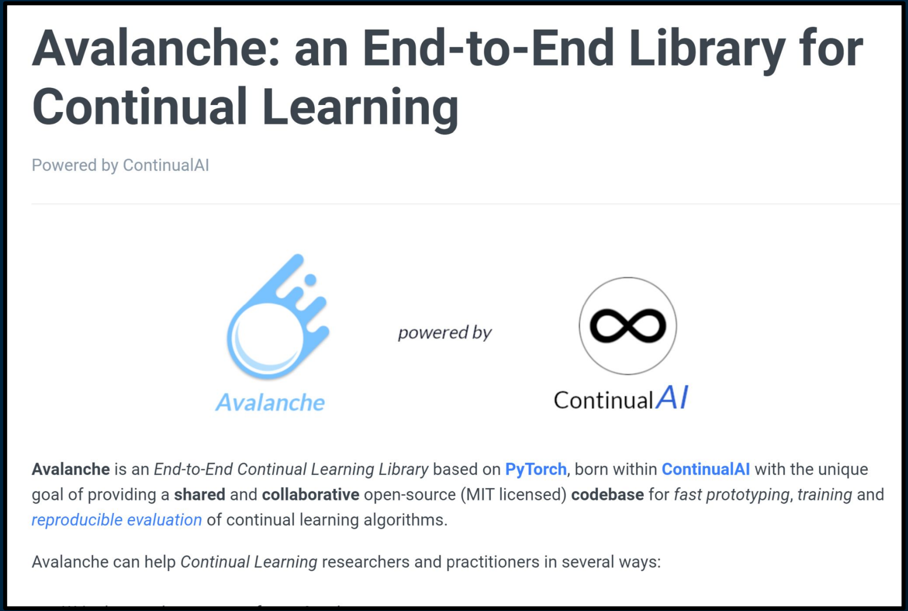


https://gantry.io

Continual Learning Tools

Avalanche: an End-to-End Library for Continual Learning

- **Powered** by the **ContinualAI non-profit**
- **W&B Best Library Award** at CLVision
- **Avalanche Dev Day** and **Beta Release**
- **Avalanche-RL** and **Reproducible-CL**: towards and **Avalanche Ecosystem**



Avalanche: an End-to-End Library for Continual Learning**.** Lomonaco et al. CLVision Workshop at CVPR 2021.

Continuum: Simple management of complex continual learning scenarios

- A simple **data loading library** for Continual Learning
- Not so different from the **Avalanche Benchmarks module**
- **Slightly simpler API** at the cost of less flexibility
- Good amount of **already available Pytorch datasets**

Continuum: Simple Management of Complex Continual Learning Scenarios

pypi package 1.1.8 | build passing | code quality B | DOI 10.5281/zenodo.5513759 | docs passing | coverage 69%

Documentation link | arXiv 2102.06253 | Youtube link

A library for PyTorch's loading of datasets in the field of Continual Learning

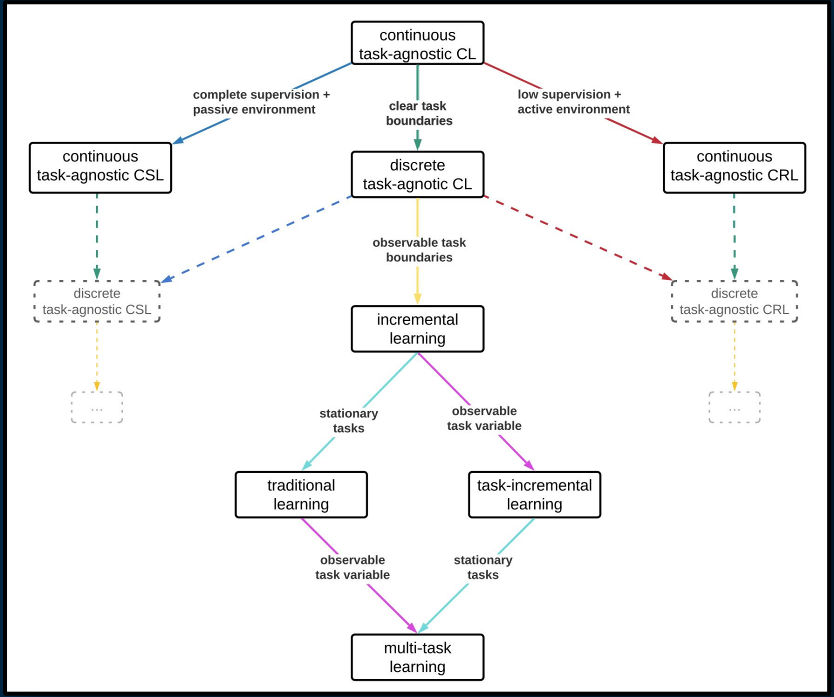
Aka Continual Learning, Lifelong-Learning, Incremental Learning, etc.

Read the documentation.
Test Continuum on Colab !

Continuum: Simple management of complex continual learning scenarios. Douillard et al. CLVision Workshop at CVPR 2021.

Sequoia: A Software Framework to Unify Continual Learning Research

- Quite **impressive engineering effort** on the grand goal of unify continual learning research in an overall hierarchy
- It builds on top of **Continuum**, **Avalanche**, **Stable-baselines** and more to provide a unique playground for Continual Learning with different levels of supervision.
- Quite different from Avalanche in terms of philosophy: more of a **top down approach**



__Sequoia: A Software Framework to Unify Continual Learning Research__. Normandin et al., CLVision Workshop at CVPR 2021.

CL-Gym: Full-Featured PyTorch Library for Continual Learning

- Similar **all-in-one library** concept as Avalanche
- Nice integration with **Pytorch Lighting**
- **Not clear if it is still maintained**

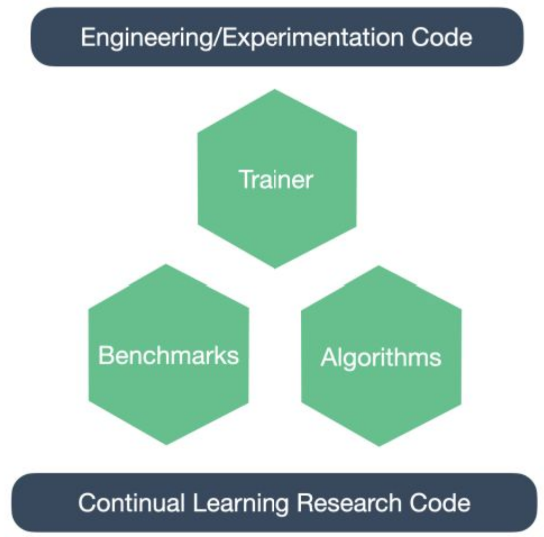


Figure 1. Main Components of *CL-Gym*

CL-Gym: Full-Featured PyTorch Library for Continual Learning. Mirzadeh et al, CLVision Workshop at CVPR 2021.

Other Interesting Software & Tools

- **Reproducible-CL**: a collection of experiments based on Avalanche for reproducing seminal papers results.
- **Avalanche-RL, Continual Habitat-Lab**: RL extension based on Avalanche, integrating the Habitat Lab 3D and Gym environments
- **Stable Baselines**: reference repo for main RL algorithms implementation
- **Meta-World**: a benchmark and evaluation suit for Meta RL
- **Continual World**: a benchmark and evaluation suit for Continual RL
- **Continual RL**: a more comprehensive continual reinforcement learning toolkit
- *...many others!*

...click on the links above to get the the respective GitHub repo!

Next:

Frontiers in Continual Learning

Do you have any questions?

vincenzo.lomonaco@unipi.it
vincenzolomonaco.com
University of Pisa

THANKS

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik