Tiny Neural Deep Clustering: An Unsupervised Approach for Continual Machine Learning on the Edge

Andrea Albanese[1], Matteo Nardello[1], and Davide Brunelli[1]

[1]Department of Industrial Engineering, University of Trento, Trento, Italy.

Continual Machine Learning (CML)

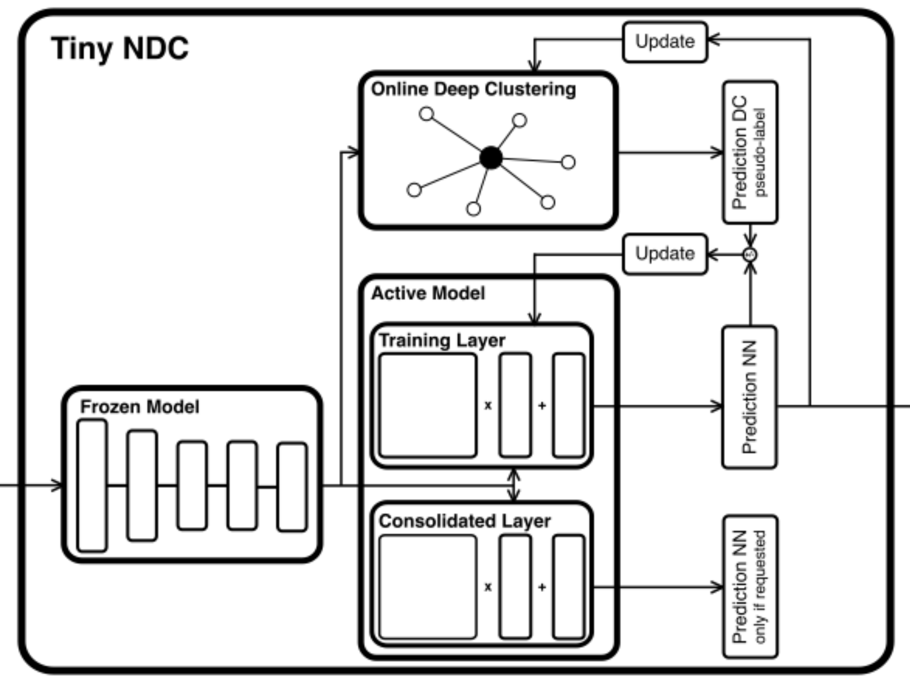
CML permits **automatic updates** of ML systems during inference.

- Overcome the context drift problem.
- Avoid system maintenance due to model re-training.
- Learn new patterns and classes.
- Adapt to environment and context changes.

However, CML systems are challenging:

- Catastrophic forgetting.
- Computationally demanding for tinyML systems.

Tiny Neural Deep Clustering (TinyNDC)



TinyNDC is a **fast** and **light** architecture that implements **unsupervised CML** on **microcontrollers**.

It is composed of three main components:

- The **Frozen model.**
- The **Active model**.
- The **Online deep clustering**.

State of the Art

Current state-of-the-art CML strategies (e.g., CWR [1], LWF [2], and tinyOL [3]) well adapt to new data.

→ **LIMIT**: they need a ground truth for carrying out the backpropagation.

Other works overcome this limitation:

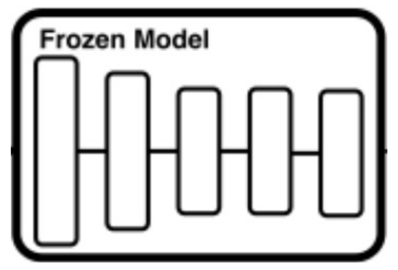
- "Online Deep Clustering" [4] implements semi-unsupervised CML by combining clustering and deep learning.
- "Unsupervised Continual Learning for Gradually Varying Domains" [5] implements clustering for classifying data in an unsupervised setting.

→ **LIMIT:** no embedded implementation.

Only the work "On-device training under 256kb memory" [6] presents an embedded implementation but in a supervised setting.

What is missing is an embedded solution for implementing unsupervised CML to get closer to real-world exploitation.

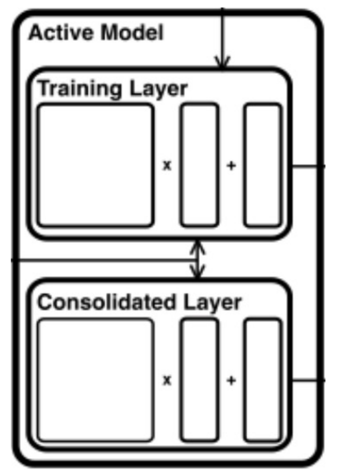
Frozen model



The **static part** of the system which is not updated during runtime. It can be **any NN truncated in the last layer** (e.g., the Softmax classifier).

- Its parameters are fixed after training and cannot be trained in real time.
- Its objective is to extract features from incoming data.

Active Model

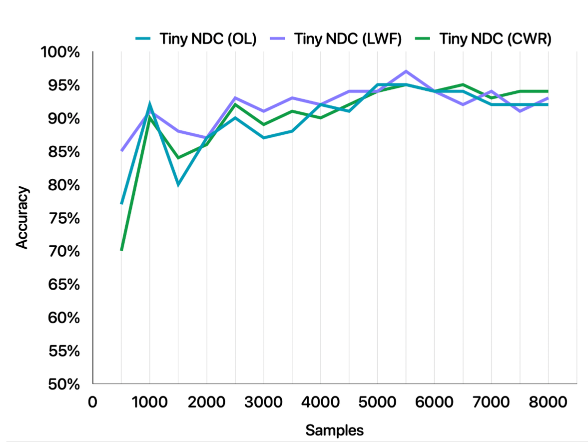


The **dynamic part** of the system which is updated during runtime. It is composed of two sub-modules, namely the **consolidated layer** and the **training layer**.

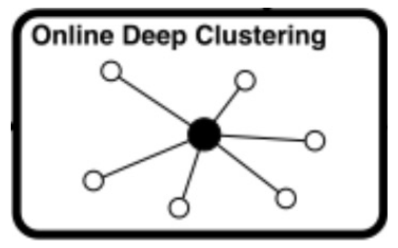
- The **consolidated layer** acts as a memory to not forget the previous knowledge.
- The **training layer** is used for domain adaptation.

The active model can **update** its **weights and biases** and **add new classes** if the current data is not associated with an already known class.

→ The selected update policy is **CWR** [1].



Online deep clustering

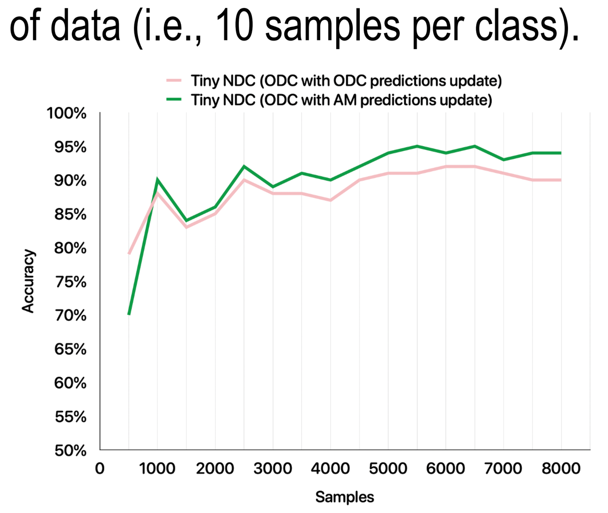


ACCURACY (%)	
1 labeled sample	75.3%±1.2%
10 labeled samples	92.3%±1.2%
50 labeled samples	93.2%±1.2%
100 labeled samples	93.5%±1.2%

- A custom algorithm that uses L2-norm to group data.

It is fed with the output of the frozen model and produces the **pseudo-label estimation**.

- The pseudo-label is fundamental for updating the active model.
- The clustering can update its centroids by using the prediction of the active model.
- The clustering's centroids need to be initialized with a small amount of data (i.e., 10 samples per class).



Case study

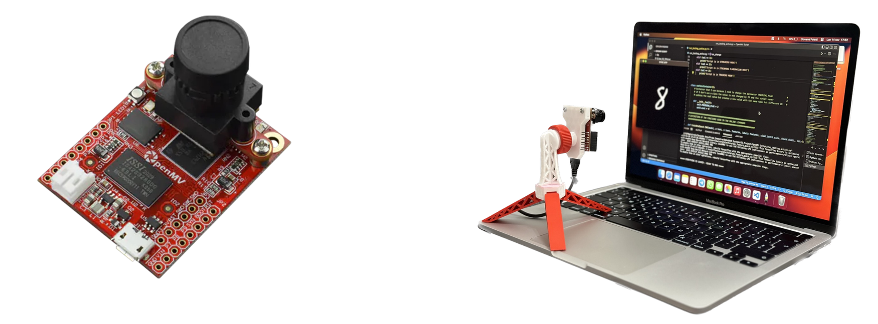
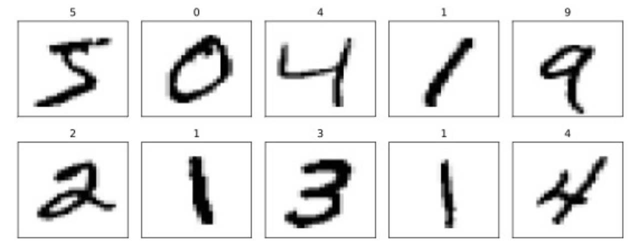


Image classification with **MNIST** dataset running on OpenMV Cam H7 Plus.

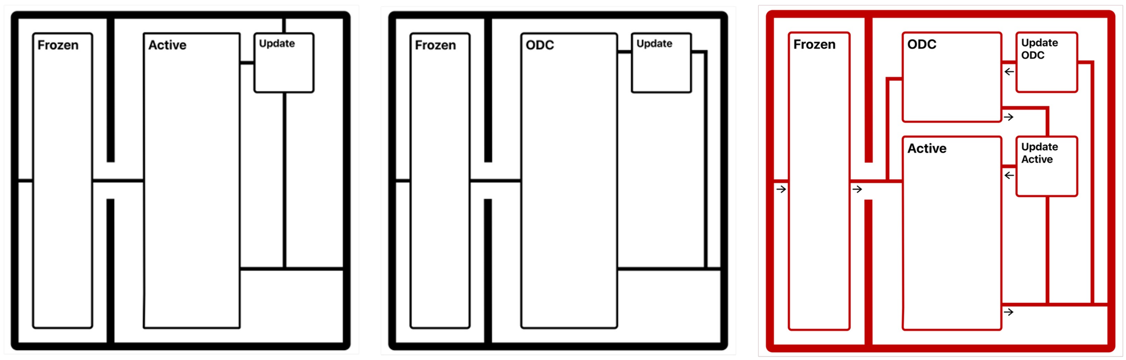
- Digits from **0 to 5** are used to train the **frozen model**.
- Digits from **6 to 9** are added as **new classes**.



- 35000 samples for training the frozen model.
- 100 samples for cluster initialization (10 per class).
- 8000 samples for testing the CML functionalities.

Experimental results

TinyNDC is compared with the same system in a supervised setting (**Supervised CML**) (i.e., without the pseudo-label estimation with clustering), and with the clustering algorithm acting as an unsupervised CML system (**Unsupervised CML**).



Supervised CML
Frozen + Active

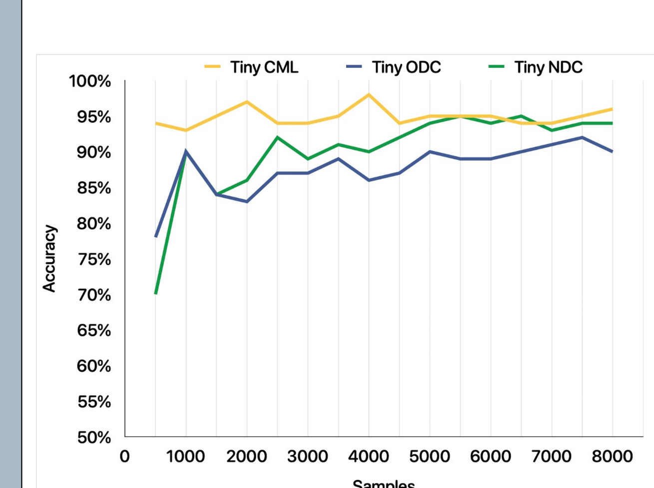
Unsupervised CML
Frozen + ODC

TinyNDC
Frozen + ODC + Active

Tests are performed with 8000 samples and scores are computed with the last 1000 samples.

1. Monitoring the degradation of performance of tinyNDC wrt supervised CML and unsupervised CML.

Algorithm	Accuracy	Precision	Recall	F-score
Supervised CML	94.3%±1.2%	95.0%±1.2%	94.0%±1.2%	94.0%±1.2%
Unsupervised CML (clustering)	90.7%±0.6%	91.0%±0.6%	91.0%±0.6%	91.0%±0.6%
Tiny NDC	92.3%±1.2%	93.0%±1.2%	92.0%±1.2%	92.0%±1.2%

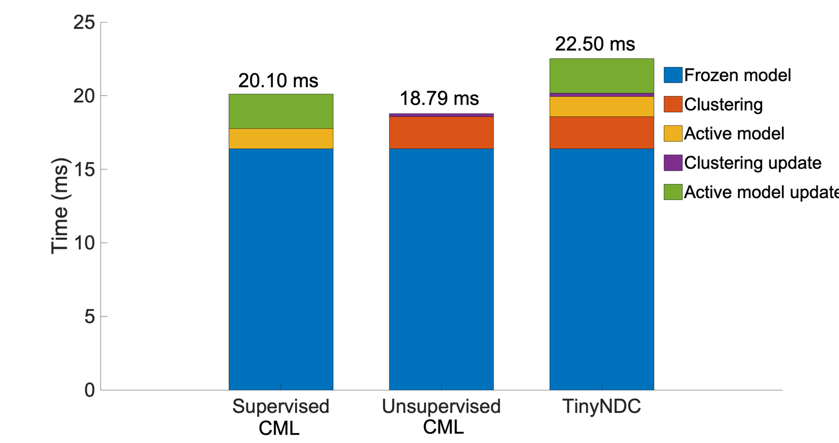


- Even though tinyNDC works in a challenging scenario, its performance is comparable with supervised CML.
- The poor performance of unsupervised CML confirms the effectiveness of tinyNDC in combining clustering with DL.

Experimental results

2. Measure the execution time of each task to ensure real-time capability and possible employment in real-world applications.

Task	Supervised CML (ms)	Unsupervised CML (clustering) (ms)	Tiny NDC (ms)
Frozen model	16.39	16.39	16.39
Clustering	-	2.18	2.18
Active model	1.38	-	1.38
Clustering update	-	0.22	0.22
Active model update	2.34	-	2.34
TOTAL	**20.10**	**18.79**	**22.50**



- TinyNDC needs more time than the other strategies to process one sample and perform the model update.
- Even though TinyNDC uses a combination of clustering and DL, the processing complexity is not increased considerably.

Conclusion

TinyNDC implements **CML** in an **unsupervised** setting.

This research proves:

- The real-time execution of such a system in an embedded device is feasible.
- The light and fast performance of the system, even though the high complexity due to the pseudo-label estimation.

The system reaches:

- **99.3%** of learning accuracy.
- Execution at **43 FPS**.

Future improvements will include:

- Update of the inner layers.
- Add clues for new classes.

References

[1] D. Maltoni and V. Lomonaco, "Continuous learning in single-incremental-task scenarios," Neural Networks, vol. 116, pp. 56-73, 2019.

[2] F. Zenke, B. Poole and S. Ganguli, "Continual learning through synaptic intelligence," International Conference on Machine Learning, pp. 3987-3995, 2017.

[3] H. Ren, D. Anicic and T. A. Runkler, "Tinyol: Tinyml with online-learning on microcontrollers," 2021 international joint conference on neural networks (IJCNN), pp. 1-8, 2021.

[4] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong and C. C. Loy, "Online deep clustering for unsupervised representation learning," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 6688-6697, 2020.

[5] A. M. N. Taufique, C. S. Jahan and A. Savakis, "Unsupervised continual learning for gradually varying domains," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3740-3750, 2022.

[6] J. a. Z. L. a. C. W.-M. a. W. W.-C. Lin, C. Gan and S. Han, "On-device training under 256kb memory," arXiv preprint arXiv:2206.15472, 2022.