Memory Reduction and Training Acceleration of Neural Networks for Tiny Machine Learning

Evangelia Fragkou and Dimitrios Katsaros
Department of Electrical and Computer Engineering
University of Thessaly, Volos

Abstract

Deep Learning is an attractive means in order to create intelligent systems and application, running on edge devices. However, due to the lack of memory capacity (approximately some MB or KB), these devices, such as embedding systems or IoT units, aren't able to support on-device learning, they can only run inference tasks. So, the problem that we have to cope with the context of Tiny Machine Learning (TinyML) is the following: "How can we reduce the size of a model in order to be deployed and implement on-line training on these devices, without sacrificing accuracy". In this article, we describe our current efforts towards achieving these goals.

Introduction

Deep learning (DL) schemes for resource-starving devices, or in other words, TinyML is an emerging research field that aims at optimizing both software tools and hardware capabilities, in order to fit to the new needs. Although it is a new research and development area, there already exist great contributions in the literature. Many pruning techniques have been surveyed in [5], with the goal of lessening the trainable parameters of the network and hence reduce its memory footprint, without sacrificing its accuracy. While most of the articles, presented in literature to address the problem of accelerating neural network training focus on implementing pruning techniques after the model being converged, our approach aims at enhancing the training time of a model during its training procedure. The only prior work, found in literature is the one presented in [4], in which the SET algorithm prunes a specific number of close-to-zero weights in a Mutli-Layer Perceptron (MLP) and restores them randomly, in every epoch. Based on the true topology of brain neurons which are scale-free [1], we investigate both network science concepts and algorithmic techniques in order to advance the ideas on neural topology pruning and thus reduce both the memory overhead of the model and its training time.

Neural Topology Pruning Algorithms

1. Brain-inspired topology pruning

We got inspiration by the way synapses of human brain are structured and by the fact that only a specific amount of these connections are important for us to keep. In addition, we observe how big networks are scaled and how they distribute information among their nodes, like the internet to maximize their performance. So, network science theories like the scale-free, small - world networks can be useful in neural networks design, too. They initialize the weights of a sparse multi-layer perceptron (MLP) in a way that they create an erdos-renyi network and after the training phase, a scale-free network is constructed. A standard number (~30%) of weights with close-to-zero values is deleted and the same number of deleted connections is restored, randomly with random weights in the network, in every epoch. In our method, we examine the way network performance is affected by changing the way the neural network is being constructed before and after the training procedure. We created five variations of scale-free and small-world network weights distributions, called SF2SFrand, SF2SFba, SF2SF(5), SF2SW, SW2SW, respectively, given as input to a feed forward neural network and after it is back-propagated, a scale-free or small-world network is constructed.

Neural Topology Pruning Algorithms

We evaluated the competitors against five datasets used in [4]. Figure 1 and Figure 2 depict the obtained results. SF2SFrand is the algorithm that outperforms both baseline method SET and the rest of our implementations not only in the accuracy in most of the cases, but also in training time.

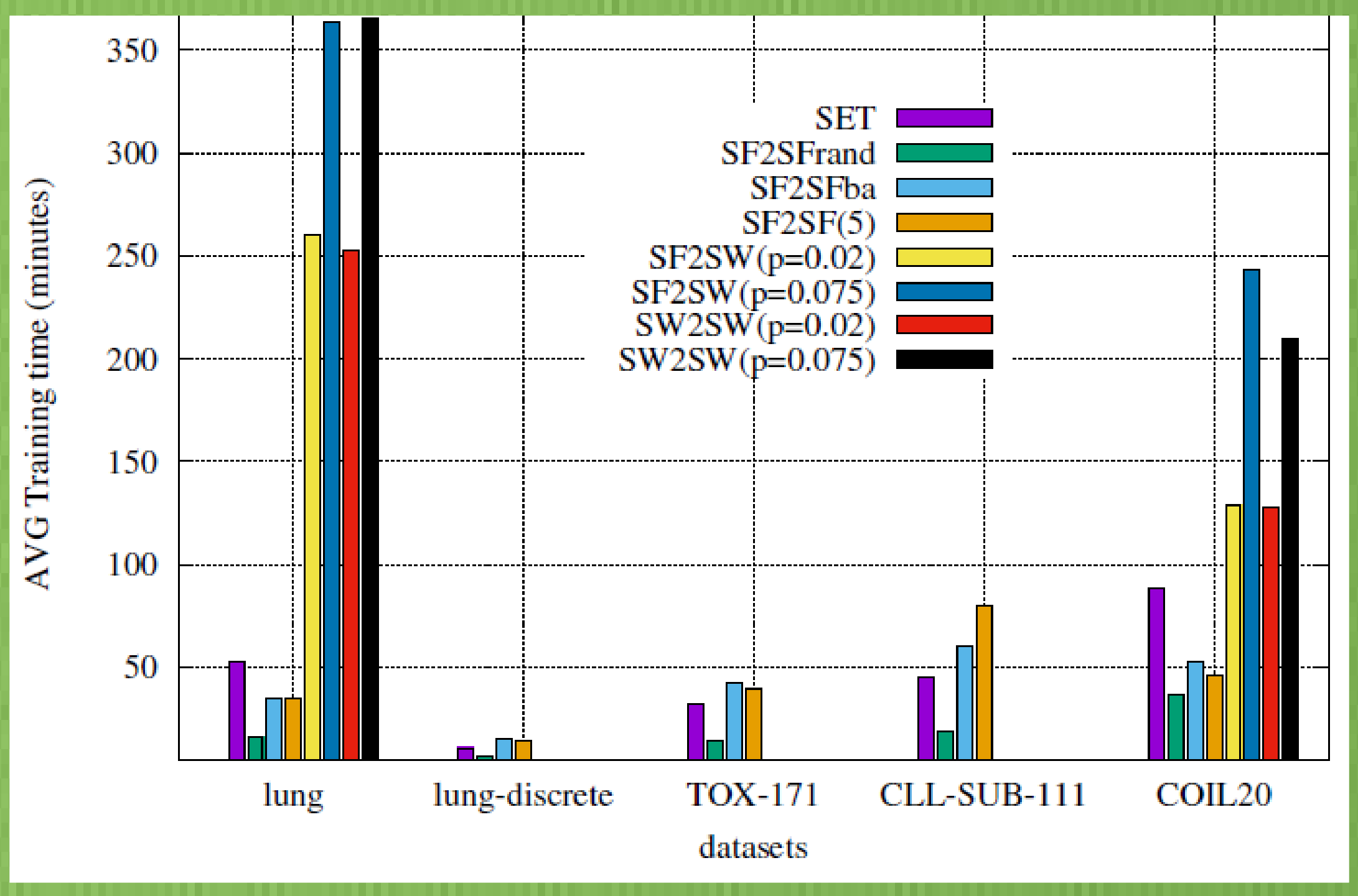


Figure 1: Average training time for convergence.

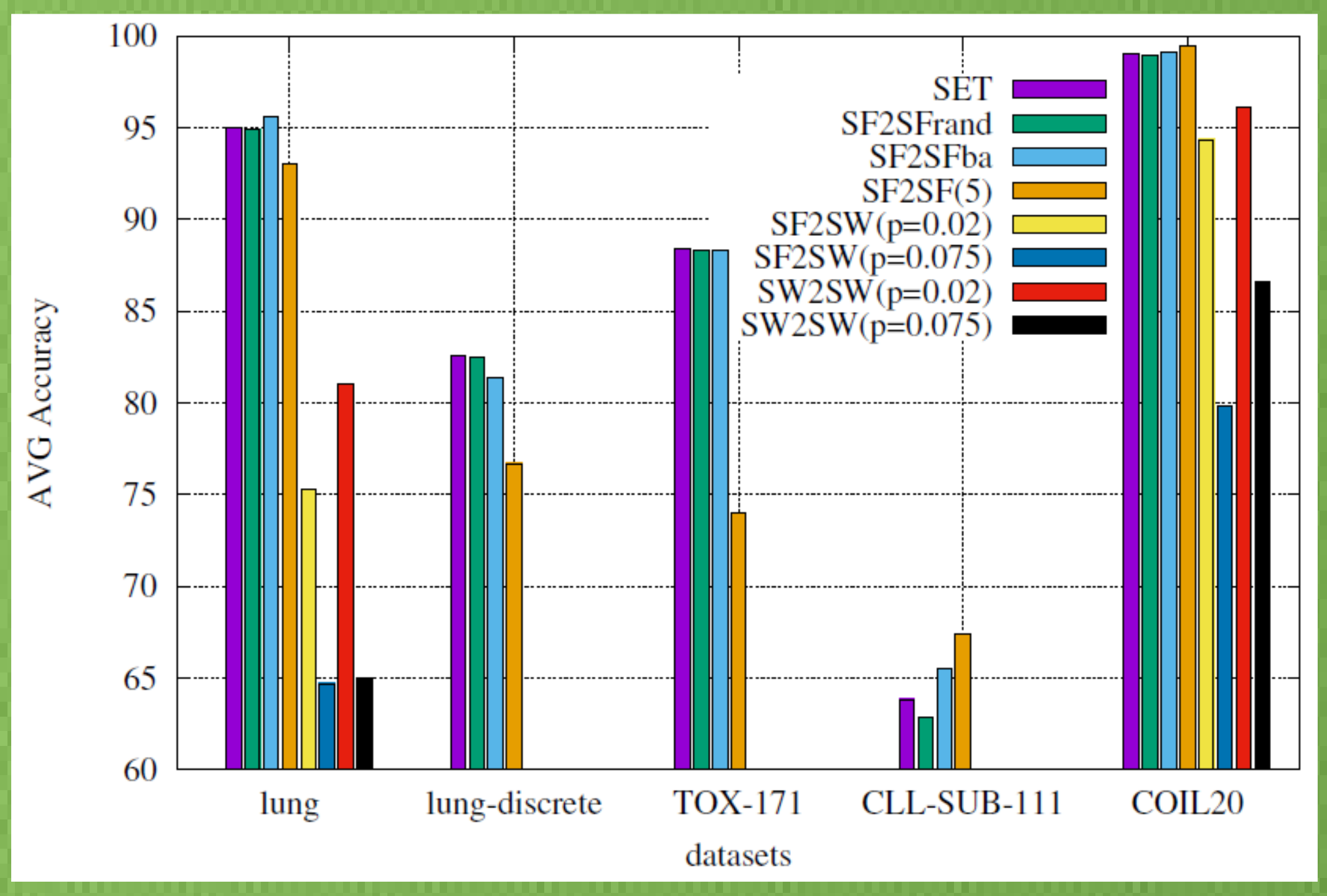
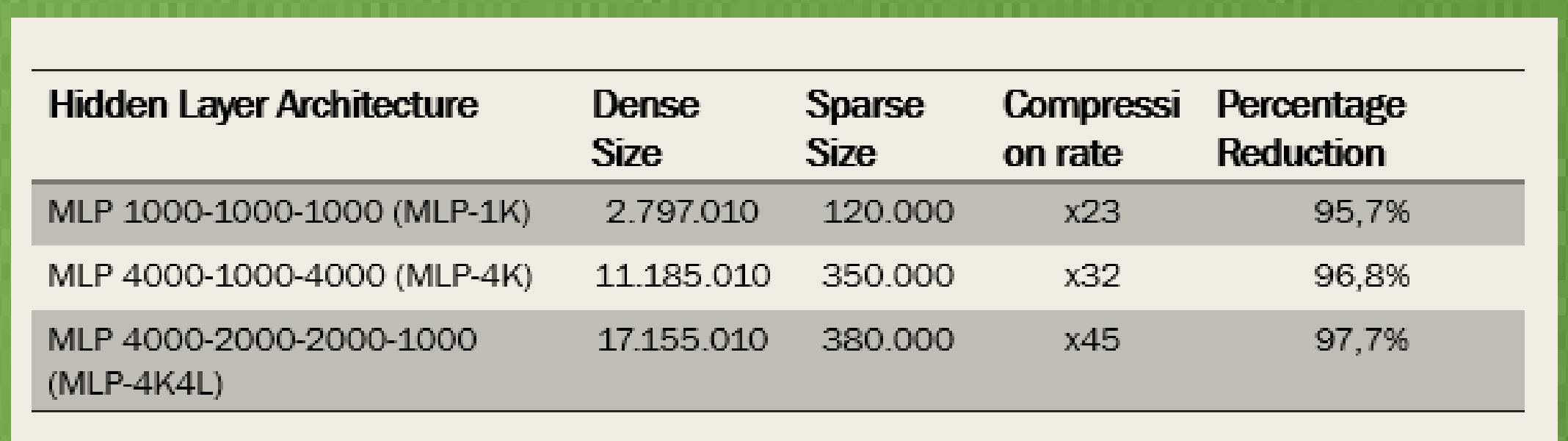


Figure 2: Average accuracy after convergence.

2. Parameterized topology pruning

In the work described above, we created different variations of network science concepts and in every epoch we modify a specific amount of connections among neurons of every layer. In our second approach, we implemented dynamic deletion / restoring of network linkages, modifying a parameter (z), which is the one responsible for the number of connections being pruned. We created three different algorithms in which z parameter changes with the ways, following: a) Linear Decreasing Variation (LDV), b) Oscillating Variation (OSV), and c) Exponential Decay (EXD). We conducted experiments with 3 different topologies and we used the Fashion MNIST dataset. Figures 3, 4 and 5 depict the obtained results in terms of the number of parameters (weights) i.e., memory overhead and accuracy. All methods presented, achieved high percentage reduction in the number of pruned parameters, while EXD method is the clear winner, in most of the cases, regarding accuracy.



Hidden Layer Architecture	Dense Size	Sparse Size	Compressi on rate	Percentage Reduction
MLP 1000-1000-1000 (MLP-1K)	2.797.010	120.000	x23	95,7%
MLP 4000-1000-4000 (MLP-4K)	11.185.010	350.000	x32	96,8%
MLP 4000-2000-2000-1000 (MLP-4K4L)	17.155.010	380.000	x45	97,7%

Figure 3: Parameter reduction between the dense MLP and the sparse methods.

Evaluation of Algorithms

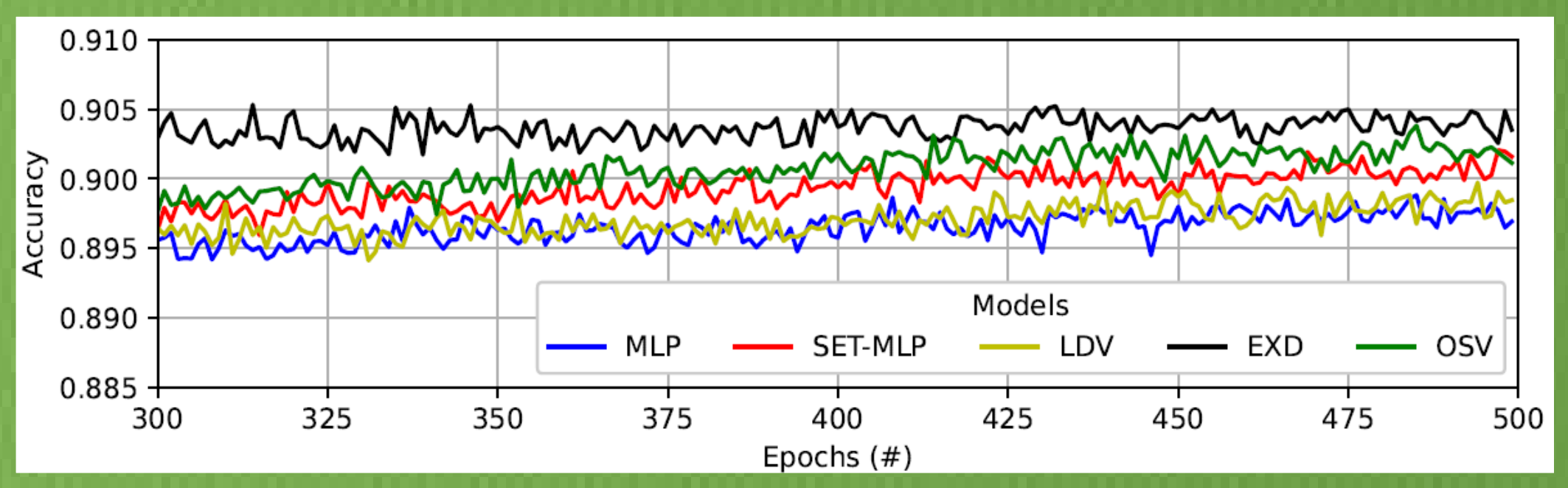


Figure 4: Competitors' accuracy based on the following network architecture: MLP 1000-1000-1000.

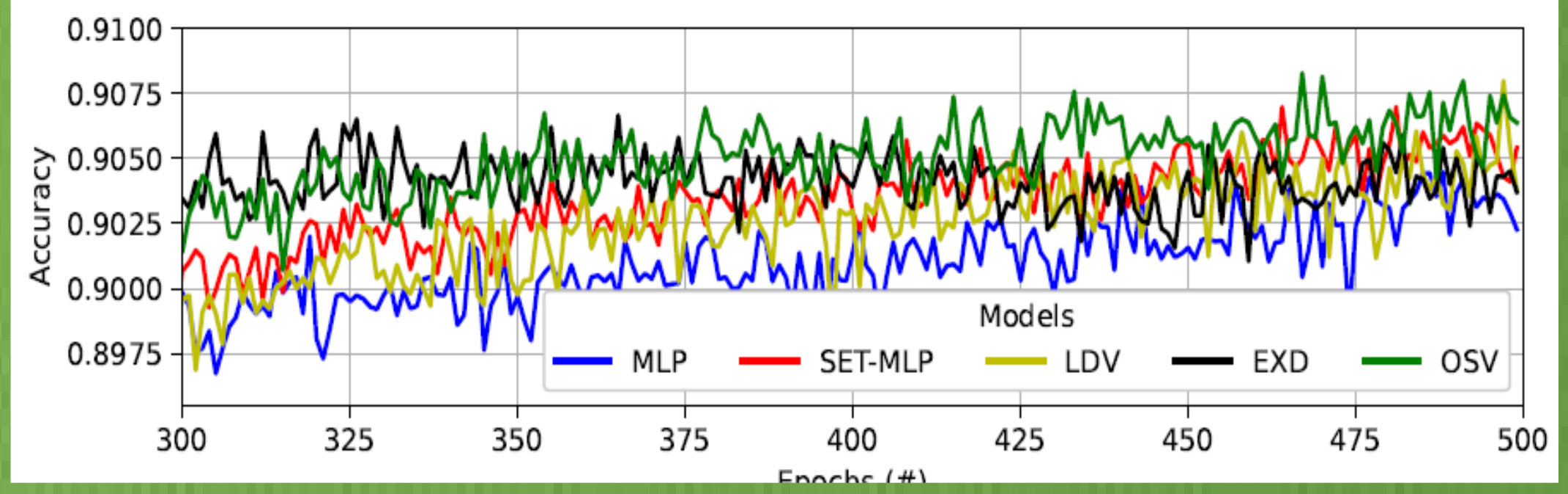


Figure 5: Competitors' accuracy based on the following network architecture: MLP 4000-2000-2000-1000.

Conclusions

In this poster, we briefly describe our current efforts in developing methods for pruning the topology of neural networks in order to boost their performance in terms of memory requirements and speedup their training time. These two requirements lie at the heart of tiny machine learning applications needs in order to make resource-starving devices such as sensors and smart phones able to run modern deep learning algorithms, e.g., federated learning over wireless networks. Our future efforts focus on developing transfer learning algorithms for tiny devices by algorithmically selecting which sets of weights to freeze and under what circumstances, while training the rest of the weights.

References

[1] E. Bullmore and O. Sporns, 2009. Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews on Neuroscience, 10*, 186–198.

[2] A. Chouliaras, E. Fragkou and D. Katsaros, 2021. Feed forward Neural network sparsification with dynamic pruning. In *Proceedings of the Panhellenic Conference on Informatics (PCI'21)*, ACM Press, Volos, Greece, 12-17.

[3] E. Fragkou, M. Koultouki and D. Katsaros. 2022. Model reduction of feed forward neural networks for resource-constrained devices. Under review for journal publication.

[4] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibesce, and A. Liotta. 2018. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications 9*.

[5] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden and A. Peste, 2021. Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research 23*, 1-124.

Acknowledgements

The research work is supported by the Hellenic Foundation for Research and Innovation (HFRI) under the 3rd Call for HFRI PhD Fellowships (Fellowship Number: 5631).