NeuroFabric: A Priori Sparsity for Training on the Edge

Mihailo Isakov and Michel A. Kinsy
Boston University
{mihailo, mkinsy}@bu.edu

***Abstract*— Many embedded devices are memory-limited, both in terms of on-chip memory and RAM. This leads to difficulties when training or running deep learning (DL) models on the edge. Pruning and quantization can reduce model size, but can only be applied *after training*. In this work we propose a priori pruning, where networks are sparse from the very start of training. By carefully picking the connection topology, we can retain the accuracy of dense networks, while significantly decreasing the model size. This can allow networks to fit into on-chip memory, enabling a greater variety of devices to train and run DL models, opening up new venues for federated learning.**

I. Training on the Edge

Many embedded devices without or with small external memories are unable to train or run deep learning models due to their large size. The nature of DL algorithms often makes caches useless, and requires frequently moving data on and off chip. RAM memory then becomes the main power consumer for embedded devices. The core of the problem lies in the large size of DL models, where both the power required to store models and to process them prevents edge devices from using DL. DL models can be heavily compressed using pruning and quantization [1], but this can only be applied *after training*. While a lot of effort has been pushed into making edge inference efficient, training models is still restricted to the datacenter. The inability of edge devices to train models, either alone or as meshes of devices, prohibits the developemnt of new applications. In order to allow training on the edge, models must be shrunk down to the bare minimum that still allows these algorithms to outperform alternatives such as decision trees, SVMs, etc. Ideally, these models should be small enough to fit into the on-chip memory of edge devices, therefore requiring less power and allowing higher throughput.

II. NeuroFabric Approach

The core of our approach is a priori sparsity - sparsity where the topology of the connections is determined *before* training. This is beneficial for several reasons: (1) the model size is reduced allowing the model to fit in on-chip memory, (2) the smaller model requires less processing power, (3) the network topology can be stored in such a way that sparse memory access patterns are linear, and (4) the network-on-chip (NoC) routing can be designed ahead of time so no processing units straggle the system. In Figure 1 we see how a conventional network is converted to an a priori sparse one. Dense layers are broken down into multiple sparse layers we call *sparse cascades*. This has the benefit of reducing the number of parameters, while maintaining full connectivity between all inputs and outputs.

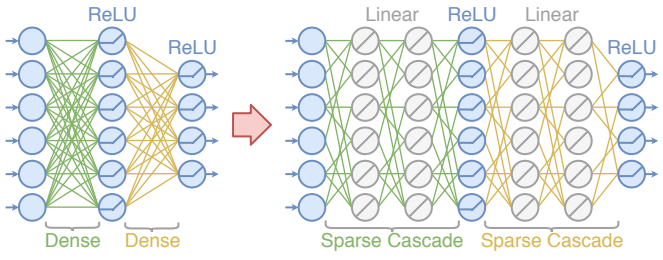

Fig. 1. A 2-layer dense network replaced with 2 sparse cascades. Cascades use linear activations in hidden and original activations in output layers.

In this work we explore why certain topologies (i.e. shallower, denser ones) outperform others (e.g. deeper sparser ones). We provide a new initialization strategy for sparse networks, and give heuristics that can determine a topology quality ahead of time. This significantly speeds up topology evaluation and allows evolutionary algorithms to arrive at good networks. We show the main bottlenecks of training deep sparse networks and provide solutions that enable even sparser, smaller models to achieve comparable accuracies. Finally, we illustrate that the same method can be applied to convolutional neural networks, allowing us to compress networks such as MobileNet 10 times without an accuracy loss.

III. Hardware Architecture & Results

Our previous work presented ClosNets [2], an a priori sparse network that replaced linear layers using sparse layers with the Clos topology. These 'ClosNets' allowed the network to decrease $5\times$ in size before training without sacrificing any accuracy (Figure 2). The Clos topology was chosen because it is only 3 layers deep, and deeper, sparser topologies were difficult to train.

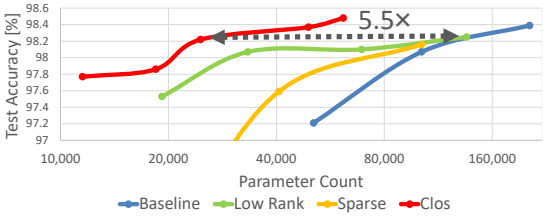

Fig. 2. MNIST accuracy vs. parameter count for dense, low-rank, a priori randomly sparse, and Clos networks.

ClosNets however had used the Clos topology that was empirically selected due to both its sparsity and an efficient torus hardware mapping, but the work did not answer why other topologies (e.g. hypercubes) would not train. This work extends ClosNets by exploring how deeper toplogies can be trained, and dives deep into the question of which intra-layer topology or 'fabric' is best suited to neural networks, both from a mathematical and an architecture perspective.

IV. Impact of A Priori Sparsity

We believe training models on the edge opens up many useful use cases. Edge training is necessary when network access is limited, user privacy is paramount, data transmission power is larger than training power, or low-latency learning is needed. However, due to a lack of low-resource training architectures, a very small amount of devices learns after being deployed. Opening up training to low-power devices allows a new class of federated learning applications to emerge, where user privacy is not sacrificed for cheap compute power.

References

[1] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," *CoRR*, vol. abs/1510.00149, 2015. [Online]. Available: http://arxiv.org/abs/1510.00149

[2] M. Isakov, A. Ehret, and M. A. Kinsy, "ClosNets: Batchless DNN Training with On-Chip a Priori Sparse Neural Topologies," in *28th International Conference on Field Programmable Logic and Applications, FPL 2018, Dublin, Ireland, August 27-31, 2018*. IEEE, 2018, pp. 55–59. [Online]. Available: https://doi.org/10.1109/FPL.2018.00017