A Hybrid Architecture for On-Device Compressive Machine Learning

Yang Li and Thomas Strohmer
Department of Mathematics
University of California, Davis
Davis, CA 95616, USA
{ly,strohmer}@math.ucdavis.edu

1 Introduction

Developing machine learning techniques that can protect the privacy of users' data is of utmost importance, as tracking and selling our digital data (often without our permission and knowledge) has become a booming business model [Zub19]. By processing the data on the device that has collected the data, we can dramatically increase the level of privacy. On-device machine offers several additional benefits, such as low latency, efficient use of network bandwidth, and more autonomy. However, many devices deployed on the "edge" have very limited memory, weak processors, and scarce energy supply. This poses the challenge of envisioning new machine learning architectures that can function properly under such dire conditions. This issue is particularly urgent with the emergence of the Internet of Things [WBSJ14].

We propose a hybrid hardware-software framework that facilitates increased privacy protection due to on-device processing and moreover has the potential to significantly reduce the computational complexity and memory requirements of on-device machine learning. In the first step, inspired by compressive sensing, data is collected in compressed form simultaneously with the sensing process. Thus this compression happens already at the hardware level during data acquisition. But unlike in compressive sensing, this compression is achieved via a projection operator that is specifically tailored to the desired machine learning task. The second step consists of a specially designed and trained deep network. Numerical simulations in image classification illustrate the viability of our method.

1.1 Prior work

Various approaches have been proposed for improving the computational cost of neural networks and for on-device deep learning. We point out that in the approaches described below, the assumptions made about the properties of these devices and their capabilities may differ for different approaches. This prior work includes new architectures to alleviate the computational burden (such as MobileNet, LCCL, SqueezeNet, etc. [IHM+16, HZC+17, DHYY17]), quantization, pruning (see e.g. [HMD15, LUW17, AANR17]), special hardware (see e.g. [Qua17, Int18, Goo18, LRY+18]), and data preprocessing ([SCC+16]. Our proposed approach is different from all the methods listed above, but can be combined with any one of them.

Other approaches towards compressing the input in a specific manner before clustering or classification can be found e.g. in [DDW+07, HS10, RP12, RRCR13, TPGV16, MMV18]. However, except for [MMV18], these papers do not taylor the compression operator to the classification task, but are rather using compression operators that are in line with classical compressive sensing.

2 Compressive Deep Learning

2.1 Outline of our approach

When we deploy an AI-equipped device in practice, we know a priori what kind of task this device is supposed to carry out. The key idea of our approach can thus be summarized as follows: We can take this knowledge into account already

in the data acquisition step itself and try to measure only the task-relevant information, thereby we significantly lower the size of the data that enter the device and thus reduce the computational burden and memory requirements of that device.

To that end we propose *compressive deep learning*, a hybrid hardware-software approach that can be summarized by two steps, see also Figure 1. First, we construct a projection operator that is specifically tailored to the desired machine learning task and which is determined by the entire training set (or a subset of the training set). This projection operator compresses the data simultaneously with the sensing process, like in standard compressive sensing [FR13]. But unlike compressive sensing, our projection operator is tailored specifically to the intended machine learning task, which therefore allows for a much more "aggressive" compression. The construction of the projection operator is of course critical and various techniques are described in the full paper. This projection will be implemented in hardware, thus the data enter the software layer of the device already in compressed form. The data acquisition/compression step is followed by a convolutional neural net (CNN) that processes the compressed data and carries out the intended task.

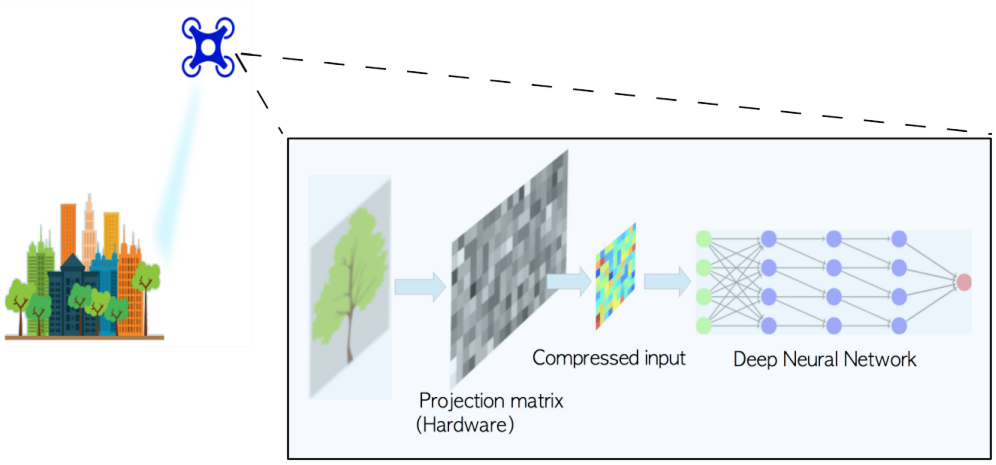

Figure 1: Schematic depiction of compressive deep learning: Data acquisition and compression are carried out simultaneously. Compression is achieved at the hardware level via a projection operator that is specifically tailored to the desired machine learning task. The so compressed data are then fed into a specially trained deep network that performs the intended task.

While our approach is applicable to a wide range of machine learning tasks, here we focus on image classification as a concrete example. We emphasize that the proposed projection/compression at the data acquisition step is completely different from standard (jpeg-type) image compression and closer in spirit to concepts from compressive sensing. Firstly, standard image compression happens at the software layer and is completely independent from the data acquisition step, while our proposed compression-while-sensing scheme is an intrinsic part of the data acquisition step. Secondly, standard image compression is designed to work for a vast range of images and independent of the task we later perform, while our compression scheme is inherently tied to the image classification task we intend to carry out.

2.2 Construction of adaptive projection operator

There are many possibilities to construct the projection operator. It is important to keep in mind that ultimately the projection/compression step is supposed to be realized in hardware. Therefore it makes sense to impose some structure on the projection operator to make it more amendable to an efficient hardware implementation. Unlike compressive sensing, we will not use a random matrix that samples the image space essentially uniformly at random, but instead we construct a projector that focuses on the regions of interest, i.e., we concentrate our measurement on those regions of the ambient space, in which the images we aim to classify are located, thereby preserving most information with a small number of measurements. To that end, principal component decomposition would suffice to construct the projection matrix. However, a typical PCA matrix is unstructured and is thus hard to implement efficiently, easily and at low cost in hardware.

Hence, we need to impose additional condition on the projection operator to be constructed. There is a range of options, but the most convenient one is arguably to consider projections with convolution structure. Convolutions are ubiquitous in signal- and image processing, they are a main ingredient of many machine learning techniques (deep learning being one of them), and they can be implemented efficiently in hardware [RG75].

We will consider two approaches to construct such a convolution-structured projection, the construction of which is described in detail in the full paper.

1. *Circulant approximate projection:* We try to find among all convolution matrices with orthogonal rows the one that is "most similar" to the PCA matrix. While this *matrix nearness problem* is non-convex, we prove in the full paper that there is a convex problem "nearby" and that this convex problem has a convenient explicit solution. This construction of the projection matrix is independent of the CNN we use for image classification.
2. Projective neural network (PNN): We construct the convolution projection by jointly optimizing the projection matrix and the CNN used for image classification. We do this by adding a "zeroth" convolution layer to our image classification CNN. The weights of this zeroth layer will give us the coefficients of the (nonunitary) convolution projection matrix. Of course, for the actual image classification we later remove this zeroth layer, since the whole point is to implement this layer in hardware. In theory this should yield a projection matrix with superior performance, because this approach jointly optimizes the projection and the classification. But due to the non-convex nature of this optimization problem, there is no guarantee that we can actually find the optimal solution.

2.3 Connection to compressive sensing

Our approach is in part inspired by ideas from compressive sensing, yet there are fundamental differences. Recall that the compressive sensing paradigm uses simultaneous sensing and compression. At the core of compressed sensing lies the discovery that it is possible to reconstruct a sparse signal exactly from an underdetermined linear system of equations and that this can be done in a computationally efficient manner via ℓ_1-minimization, cf. [FR13]. Compressive sensing consists of two parts: (i) The sensing step, which simultaneously compresses the signal. This step is usually implemented (mostly) in hardware. (ii) The signal reconstruction step via carefully designed algorithms (thus this is done by software).

Compressed sensing aligns with a few of our objectives, however, they also differ in the following crucial ways: (i) *Adaptivity:* Standard compressive sensing is not adaptive. It considers all possible sparse signals under certain representations. For different data sets, the fundamental assumptions are the same. This assumption is likely too weak for a specific image data set, where only images with certain characteristics are included. For image classification, this is usually the case. Statistical information in the data set may be exploited to achieve better results. (ii) *Exact reconstruction:* Compressive sensing aims for exact signal reconstruction. That means enough measurements must be taken to ensure all information needed for exact recovery. Clearly, this is too stringent a constraint for image classification where only label recovery is required instead of full signal reconstruction. (iii) *Storage and processing cost:* It is cumbersome to implement random matrices often proposed in compressive sensing in hardware; only certain structured random matrices can be implemented efficiently.

2.4 Privacy protecting machine learning

As mentioned above, protecting the privacy of a user's data is (or should be) of utmost importance in many applications. Our approach aids to this goal in two ways: (i) Privacy and security are much harder to compromise if data is processed locally instead of being sent to the cloud. Thus by processing data directly on the device, we improve privacy significantly. Our approach aims at making on-device processing possible even for devices that are very limited in terms of computing power and memory, and thus it may be instrumental in bringing privacy protection to the Internet of Things. (ii) Since data are already entering the device in compressive form, this provides automatically a form of low-level encryption against semi-honest agents such as giant Internet companies – hostile agents however may be able to break this encryption to some extent.

3 Numerical experiments

To demonstrate the capabilities of our methods, we test both the PNN and the circulant approximation against a few other baseline methods. We test these methods on two standard test sets, the MNIST dataset consisting of images depicting hand-written digits, and the Fashion-MNIST dataset, consisting of images of fashion products. The general workflow consists of two steps. In the preprocessing step we subject the images to the projection operator to simulate the compressive image acquisition via hardware. In the second step we feed these images into a convolutional neural network for classification. The details of the experiments are given in the full paper.

As shown in Figure 2, for both datasets, it is evident that both PNN and the circulant approximation achieve higher accuracy rate than downsampling and random convolution, especially when the data are heavily compressed. In most

Stride	Dimension	Compression
1	28×28	1.00
2	14×14	4.00
3	10×10	7.84
4	7×7	16.00
5	6×6	21.78
6	5×5	31.36

Table 1: Relation between the stride and the compression rate. The dimension of the raw inputs is 28×28. After applying one of the preprocessing method with a certain stride, the dimension of the data becomes smaller. The compression is the ratio between the number of pixels in the the raw data and that of the processed data.

cases, the PNN method works slightly better than the circulant approximation. However, the circulant approximation method exhibits its ability to retain high accuracy when pushing to more extreme compression rate on MNIST. Since PNN is a relaxation of the circulant approximation method, the global optimum of the former is always no worse than the latter. What we observe in MNIST is a result of the training process, which has no guarantee for global optimum of the neural network. One can also see that using a subsampled random convolution matrix as often used in compressive sensing (see e.g. [Rom09, FR13]) gives much worse performance than the proposed techniques.

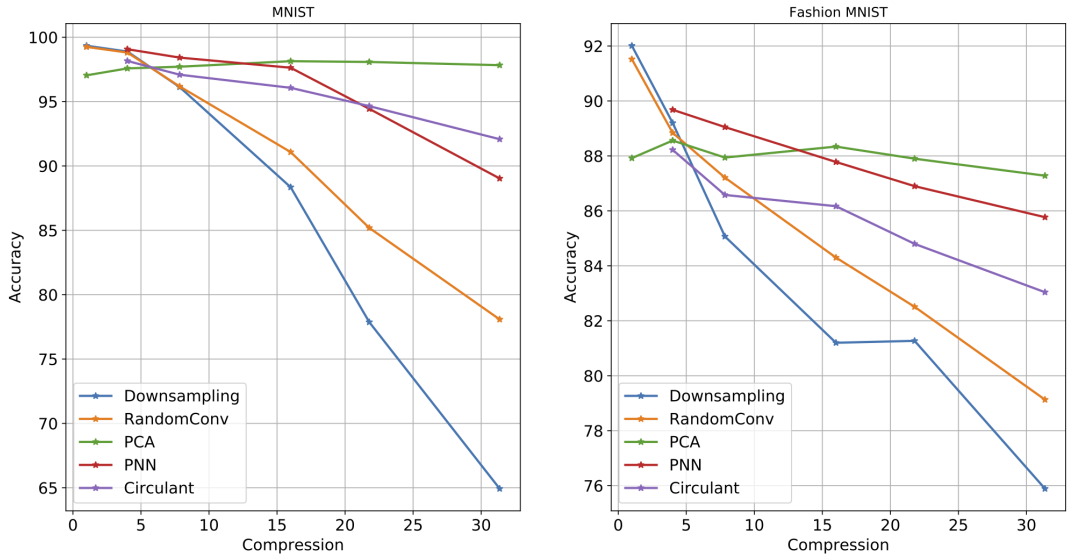

Figure 2: Accuracy rates of compressive deep learning with various choices for the projection matrix, tested on MNIST and on Fashion-MNIST. The x-axis represents the compression rate of the input (this is determined by the strides). The relation between the strides and the compression is given in Table 1.

Another interesting result is that the PCA method seems not to be affected by extreme compression rates but rather benefits from them. This is probably because only the coefficients of the leading PCA components have high signal-to-noise ratio, and the rest are mostly noise. Therefore, the PCA method performs better simply by discarding the noisy coefficients. For the purpose of our work, the PCA method cannot be compared with the other methods directly since *the PCA matrix is not a convolution and cannot be implemented by hardware*.

Remark: The investigations discussed in this note are just a first step. Many problems remain, new questions arise, and more detailed simulations are necessary to assess the full potential of this compressive hybrid framework for on-device machine learning.

Acknowledgments

The authors want to thank Donald Pinckney for discussions and initial simulations related to the topic of this paper. The authors acknowledge partial support from NSF via grant DMS 1620455 and from NGA and NSF via grant DMS 1737943.

References

[AANR17] Alireza Aghasi, Afshin Abdi, Nam Nguyen, and Justin Romberg. Net-trim: Convex pruning of deep neural networks with performance guarantee. In *Advances in Neural Information Processing Systems*, pages 3177–3186, 2017.

[CHS96] John H. Conway, Ronald H. Hardin, and Neil JA Sloane. Packing lines, planes, etc.: Packings in grassmannian spaces. *Experimental mathematics*, 5(2):139–159, 1996.

[Dav79] Philip J. Davis. *Circulant matrices*. John Wiley & Sons, 1979.

[DDW+07] Mark A Davenport, Marco F Duarte, Michael B Wakin, Jason N Laska, Dharmpal Takhar, Kevin F Kelly, and Richard G Baraniuk. The smashed filter for compressive classification and target recognition. In *Computational Imaging V*, volume 6498, page 64980H. International Society for Optics and Photonics, 2007.

[DHYY17] Xuanyi Dong, Junshi Huang, Yi Yang, and Shuicheng Yan. More is less: A more complicated network with less inference complexity. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[DP17] M. Deisher and A. Polonski. Implementation of efficient, low power deep neural networks on next-generation intel client platforms. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6590–6591, March 2017.

[FR13] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Springer, Basel, 2013.

[Goo18] Google. Edge TPU, 2018. https://cloud.google.com/edge-tpu.

[HMD15] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[HS10] Blake Hunter and Thomas Strohmer. Compressive spectral clustering. In *AIP Conference Proceedings*, volume 1281(1), pages 1720–1722. AIP, 2010.

[HZC+17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[IHM+16] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[Int18] Intel. Hardware, 2018. https://ai.intel.com/hardware.

[LRY+18] Xing Lin, Yair Rivenson, Nezih T Yardimci, Muhammed Veli, Yi Luo, Mona Jarrahi, and Aydogan Ozcan. All-optical machine learning using diffractive deep neural networks. *Science*, 361(6406):1004–1008, 2018.

[LUW17] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. In *Advances in Neural Information Processing Systems*, pages 3288–3298, 2017.

[MMV18] Culver McWhirter, Dustin G Mixon, and Soledad Villar. Squeezefit: Label-aware dimensionality reduction by semidefinite programming. *arXiv preprint arXiv:1812.02768*, 2018.

[Qua17] Qualcomm. We are making on-device AI ubiquitous, 2017. https://www.qualcomm.com/news/onq/2017/08/16/we-are-making-device-ai-ubiquitous.

[RG75] Lawrence R Rabiner and Bernard Gold. *Theory and application of digital signal processing*. Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975.

[Rom09] Justin Romberg. Compressive sensing by random convolution. *SIAM Journal on Imaging Sciences*, 2(4):1098–1128, 2009.

[RP12] Andrzej Ruta and Fatih Porikli. Compressive clustering of high-dimensional data. In *2012 11th International Conference on Machine Learning and Applications*, volume 1, pages 380–385. IEEE, 2012.

[RRCR13] Hugo Reboredo, Francesco Renna, Robert Calderbank, and Miguel RD Rodrigues. Compressive classification. In *2013 IEEE International Symposium on Information Theory*, pages 674–678. IEEE, 2013.

[SCC+16] Alaa Saade, Francesco Caltagirone, Igor Carron, Laurent Daudet, Angélique Drémeau, Sylvain Gigan, and Florent Krzakala. Random projections through multiple optical scattering: Approximating kernels at the speed of light. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6215–6219. IEEE, 2016.

[TPGV16] Nicolas Tremblay, Gilles Puy, Rémi Gribonval, and Pierre Vandergheynst. Compressive spectral clustering. In *International Conference on Machine Learning*, pages 1002–1011, 2016.

[WBSJ14] G. Wunder, H. Boche, T. Strohmer, and P. Jung. Sparse signal processing concepts for efficient 5G system design, *IEEE Access*, 3 (2015), pp. 195–208.

[Zub19] Shoshana Zuboff. *The Age of Surveillance Capitalism: The Fight for the Future at the New Frontier of Power*. PublicAffairs, 2019.