Memory-efficient Learning for Large-scale Computational Imaging

Michael Kellman [1] **Jon Tamir** [1] **Emrah Bostan** [1] **Michael Lustig** [1] **Laura Waller** [1]

Abstract

Computational imaging systems jointly design computation and hardware to retrieve information which is not traditionally accessible with standard imaging systems. Recently, critical aspects such as experimental design and image priors are optimized through deep neural networks formed by the unrolled iterations of classical physics-based reconstructions (termed physics-based networks). However, for real-world large-scale systems, computing gradients via backpropagation restricts learning due to memory limitations of graphical processing units. In this work, we propose a memory-efficient learning procedure that exploits the reversibility of the network's layers to enable data-driven design for large-scale computational imaging. We demonstrate our methods practicality on two large-scale systems: super-resolution optical microscopy and multi-channel magnetic resonance imaging.

1. Introduction

Computational imaging systems (tomographic systems, computational optics, magnetic resonance imaging, to name a few) jointly design software and hardware to retrieve information which is not traditionally accessible on standard imaging systems. Generally, such systems are characterized by how the information is encoded (forward process) and decoded (inverse problem) from the measurements. The decoding process is typically iterative in nature, alternating between enforcing data consistency and image prior knowledge. Recent work has demonstrated the ability to optimize computational imaging systems by unrolling the iterative decoding process to form a differentiable Physics-based Network (PbN) (1; 2; 3) and then relying on a dataset and training to learn the system's design parameters, *e.g.* experimental design (3; 4; 5), image prior model (1; 2; 6; 7). PbNs are constructed from the operations of reconstruction, *e.g.* proximal gradient descent algorithm. By including known structures and quantities, such as the forward model, gradient, and proximal updates, PbNs can be efficiently parameterized by only a few learnable variables, thereby enabling an efficient use of training data (6) while still retaining robustness associated with conventional physics-based inverse problems.

Training PbNs relies on gradient-based updates computed using backpropagation (an implementation of reverse-mode differentiation (8)). Most modern imaging systems seek to decode ever-larger growing quantities of information (gigabytes to terabytes) and as this grows, memory required to perform backpropagation is limited by the memory capacity of modern graphical processing units (GPUs).

Methods to save memory during backpropagation (*e.g.* forward recalculation, reverse recalculation, and checkpointing) trade off spatial and temporal complexity (8). For a PbN with N layers, standard backpropagation achieves $\mathcal{O}(N)$ temporal and spatial complexity. Forward recalculation achieves $\mathcal{O}(1)$ memory complexity, but has to recalculate unstored variables forward from the input of the network when needed, yielding $\mathcal{O}(N^2)$ temporal complexity. Forward checkpointing smoothly trades off temporal, $\mathcal{O}(NK)$, and spatial, $\mathcal{O}(N/K)$, complexity by saving variables every K layers and forward-recalculating unstored variables from the closest checkpoint.

Reverse recalculation provides a practical solution to beat the trade off between spatial vs. temporal complexity by calculating unstored variables in reverse from the output of the network, yielding $\mathcal{O}(N)$ temporal and $\mathcal{O}(1)$ spatial complexities. Recently, several reversibility schemes have been proposed for residual networks (9), learning ordinary differential equations (10), and other specialized network architectures (11; 12).

In this work, we propose a memory-efficient learning procedure for backpropagation for the PbN formed from proximal gradient descent, thereby enabling learning for many large-scale computational imaging systems. Based on the concept of invertibility and reverse recalculation, we detail how backpropagation can be performed without the need to store intermediate variables for networks composed of gradient and proximal layers. We highlight practical restrictions on the layers and introduce a hybrid scheme that combines our reverse recalculation methods with checkpointing to mitigate numerical error accumulation. Finally, we demonstrate our method's usefulness to learn the design for two

[1]Electrical Engineering and Computer Sciences, University of California, Berkeley, USA. Correspondence to: Michael Kellman <kellman@berkeley.edu>.

practical large-scale computational imaging systems: super-resolution optical microscopy (Fourier Ptychography) and multi-channel magnetic resonance imaging.

2. Background

Computational imaging systems are described by how sought information is encoded to and decoded from a set of measurements. The encoding of information, $\mathbf{x}$ into measurements, $\mathbf{y}$, is given by

$$\mathbf{y} = \mathcal{A}(\mathbf{x}) + \mathbf{n}, \tag{1}$$

where $\mathcal{A}$ is the forward model that characterizes the measurement system physics and $\mathbf{n}$ is random system noise. The forward model is a continuous process, but is often approximated by a discrete representation. The retrieval of information from a set of measurements, *i.e.* decoding, is commonly structured using an inverse problem formulation,

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} \mathcal{D}(\mathbf{x}; \mathbf{y}) + \mathcal{P}(\mathbf{x}), \tag{2}$$

where $\mathcal{D}(\cdot)$ is a data fidelity penalty and $\mathcal{P}(\cdot)$ is a prior penalty. When $\mathbf{n}$ is governed by a known noise model, the data consistency penalty can be written as the negative log-likelihood of the appropriate distribution. When $\mathcal{P}(\cdot)$ is a non-smooth prior (*e.g.* ℓ_1, total variation), proximal gradient descent (PGD) and its accelerated variants are often efficient algorithms to minimize the objective in Eq. 2 and are composed of the following alternating steps:

$$\mathbf{z}^{(k)} = \mathbf{x}^{(k)} - \alpha \nabla_{\mathbf{x}} \mathcal{D}(\mathbf{x}^{(k)}; \mathbf{y}), \tag{3}$$

$$\mathbf{x}^{(k+1)} = \texttt{prox}_{\mathcal{P}}(\mathbf{z}^{(k)}), \tag{4}$$

where α is the gradient step size, $\nabla_{\mathbf{x}}$ is the gradient operator, $\text{prox}_{\mathcal{P}}$ is a proximal function that enforces the prior (13), and $\mathbf{x}^{(k)}$ and $\mathbf{z}^{(k)}$ are intermediate variables for the k^{th} iteration.

The structure of the PbN is determined by unrolling N iterations of the optimizer to form the N layers of a network (Eq. 3 and Eq. 4 form a single layer). Specifically, the input to the network is the initialization of the optimization, $\mathbf{x}^{(0)}$, and the output is the resultant, $\mathbf{x}^{(N)}$. The learnable parameters are optimized using gradient-based methods. Common machine learning toolboxes' (*e.g.* PyTorch, Tensor Flow, Caffe) auto-differentiation functionalities are used to compute gradients for backpropagation. Auto-differentiation accomplishes this by creating a graph composed of the PbN's operations and storing intermediate variables in memory.

3. Methods

Our main contribution is to improve the spatial complexity of backpropagation for PbNs by treating the larger single graph for auto-differentiation as a series of smaller graphs. Specifically, consider a PbN, $\mathcal{F}$, composed of a sequence of layers,

$$\mathbf{x}^{(k+1)} = \mathcal{F}^{(k)}\left(\mathbf{x}^{(k)}; \theta^{(k)}\right), \tag{5}$$

where $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k+1)}$ are the k^{th} layer input and output, respectively, and $\theta^{(k)}$ are its learnable parameters. When performing reverse-mode differentiation, our method treats a PbN of N layers as N separate smaller graphs, processed one at a time, rather than as a single large graph, thereby saving a factor N in memory. As outlined in Alg. 1, we first recalculate the current layer's input, $\mathbf{x}^{(k-1)}$, from its output, $\mathbf{x}^{(k)}$, using $\mathcal{F}^{(k-1)}_{\text{inverse}}$, and then form one of the smaller graphs by recomputing the output of the layer, $\mathbf{v}^{(k)}$, from the recalculated input. To compute gradients, we then rely on auto-differentiation of each layer's smaller graph to compute the gradient of the loss, $\mathcal{L}$, with respect to $\mathbf{x}^{(k)}$ (denoted $\mathbf{q}^{(k)}$) and $\nabla_{\theta^{(k)}}\mathcal{L}$. The procedure is repeated for all N layers in reverse order.

Algorithm 1 Memory-efficient learning for physics-based networks

procedure MEMORY-EFFICIENT BACKPROPAGATION($\mathbf{x}^{(N)}, \mathbf{q}^{(N)}$)
  $k \leftarrow N$
  for $k > 0$ **do**
    $\mathbf{x}^{(k-1)} \leftarrow \mathcal{F}^{(k-1)}_{\text{inverse}}(\mathbf{x}^{(k)}; \theta^{(k-1)})$
    $\mathbf{v}^{(k)} \leftarrow \mathcal{F}^{(k-1)}(\mathbf{x}^{(k-1)}; \theta^{(k-1)})$
    $\mathbf{q}^{(k-1)} \leftarrow \frac{\partial \mathbf{v}^{(k)}}{\partial \mathbf{x}^{(k-1)}} \mathbf{q}^{(k)}$
    $\nabla_{\theta^{(k)}}\mathcal{L} \leftarrow \frac{\partial \mathbf{v}^{(k)}}{\partial \theta^{(k)}} \mathbf{q}^{(k)}$
    $k \leftarrow k - 1$
  end for
  return $\{\nabla_{\theta^{(k)}}\mathcal{L}\}_{k=0}^{N-1}$
end procedure

In order to perform the reverse-mode differentiation efficiently, we must be able to compute each layer's inverse operation, $\mathcal{F}^{(k-1)}_{\text{inverse}}$. The remainder of this section overviews the procedures to invert gradient and proximal update layers.

3.1. Inverse of gradient update layer

A common interpretation of gradient descent is as a forward Euler discretization of a continuous-time ordinary differential equation. As a consequence, the inverse of the gradient step layer (Eq. 3) can be viewed as a backward Euler step,

$$\mathbf{x}^{(k)} = \mathbf{z}^{(k)} + \alpha \nabla_{\mathbf{x}} \mathcal{D}(\mathbf{x}^{(k)}; \mathbf{y}). \tag{6}$$

This implicit equation can be solved iteratively via the backward Euler method using the fixed point algorithm (Alg. 2). Convergence is guaranteed if

$$\text{Lip}\left(\alpha \nabla_{\mathbf{x}} \mathcal{D}(\mathbf{x}; \mathbf{y})\right) < 1, \tag{7}$$

where $\text{Lip}(\cdot)$ computes the Lipschitz constant of its argument (14). In the setting when $\mathcal{D}(\mathbf{x};\mathbf{y}) = \|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2$ and $\mathbf{A}$ is linear this can be ensured if $\alpha < \frac{1}{\sigma_{max}(\mathbf{A}^H\mathbf{A})}$, where $\sigma_{max}(\cdot)$ computes the largest singular value of its argument. Finally, as given by Banach Fixed Point Theorem, the fixed point algorithm (Alg. 2) will have an exponential rate of convergence (14).

Algorithm 2 Inverse for gradient layer

procedure FIXED POINT METHOD($\mathbf{z}, T$)
  $\mathbf{x} \leftarrow \mathbf{z}$
  for $t < T$ **do**
    $\mathbf{x} \leftarrow \mathbf{z} + \alpha\nabla_{\mathbf{x}}\mathcal{D}(\mathbf{x};\mathbf{y})$
    $t \leftarrow t+1$
  end for
  return $\mathbf{x}$
end procedure

3.2. Inverse of proximal update layer

The proximal update (Eq. 4) is defined by the following optimization problem (13):

$$\text{prox}_{\mathcal{P}}(\mathbf{z}^{(k)}) = \arg\min_{\mathbf{v}} \frac{1}{2}\|\mathbf{v}-\mathbf{z}^{(k)}\|_2^2 + \mathcal{P}(\mathbf{v}). \quad (8)$$

For differentiable $\mathcal{P}(\cdot)$, the optimum of which is,

$$\mathbf{x}^{(k+1)} = \mathbf{z}^{(k)} - \nabla_{\mathbf{x}}\mathcal{P}(\mathbf{x}^{(k+1)}). \quad (9)$$

In contrast to the gradient update layer, the proximal update layer can be thought of as a backward Euler step (13). This allows its inverse to be expressed as a forward Euler step,

$$\mathbf{z}^{(k)} = \mathbf{x}^{(k+1)} + \nabla_{\mathbf{x}}\mathcal{P}(\mathbf{x}^{(k+1)}), \quad (10)$$

when the proximal function is bijective (*e.g.* prox_{ℓ_2}). If the proximal function is not bijective (*e.g.* prox_{ℓ_1}) the inversion is not straight forward. However, in many cases it is possible to substitute it with a bijective function with similar behavior.

4. Hybrid Reverse Recalculation and Checkpointing

Reverse recalculation of the unstored variables is non-exact as the operations to calculate the variables are not identical to forward calculation. The result is numerical error between the original forward and reverse calculated variables and as more iterations are unrolled, numerical error can accumulate.

To mitigate these effects, some of the intermediate variables can be stored from forward calculation, referred to as checkpoints. Memory permitting, as many checkpoints should be stored as possible to ensure accuracy while performing reverse recalculation. While most PbNs cannot afford to store all variables required for reverse-mode differentiation, it is often possible to store a few.

5. Results

5.1. Learned experimental design for super resolution optical microscopy

Standard bright-field microscopy offers a versatile system to image *in vitro* biological samples, however, is restricted to imaging either a large field of view or a high resolution. Fourier Ptychographic Microscopy (FPM) (15) is a super resolution (SR) method that can create gigapixel-scale images beating this trade off on a standard optical microscope by acquiring a series of measurements (up to hundreds) under various illumination settings on an LED array microscopy (16) and combining them via a phase retrieval based optimization. The system's dependence on many measurements inhibits its ability to image live fast-moving biology. Reducing the number of measurements is possible using linear multiplexing (17) and state of the art performance is achieved by forming a PbN and learning its experimental design (4; 3), however, is currently limited in scale due to GPU memory constraints (terabyte-scale memory is required for learning the full measurement system). With our proposed memory-efficient learning framework, we reduce the required memory to only a few gigabytes, thereby enabling the use of consumer-grade GPU hardware.

To evaluate accuracy we compare standard learning with our proposed memory-efficient learning on a problem that fits in standard GPU memory. We reproduce results in (4) where the number of measurements are reduced by a factor of 10 using 6.26GB of memory using only 0.627GB and time is only increased by a factor of 2. To perform memory-efficient learning, we set $T = 4$ and checkpoint every 10 unrolled iterations. The testing loss between our method and standard learning are comparable (Fig. 1a). In addition, we qualitatively highlight equivalence of the two methods, displaying SR reconstructions with learned design using standard (Fig. 1d) and memory-efficient (Fig. 1e) methods. For relative comparison, we display a single low resolution measurement (Fig. 1b) and the ground truth SR reconstruction using all measurements (Fig. 1c).

5.2. Learned priors for multi-channel MRI

MRI is a powerful Fourier-based medical imaging modality that non-invasively captures rich biophysical information without ionizing radiation. Since MRI acquisition time is directly proportional to the number of acquired measurements, reducing measurements leads to immediate impact on patient throughput and enables capturing fast-changing

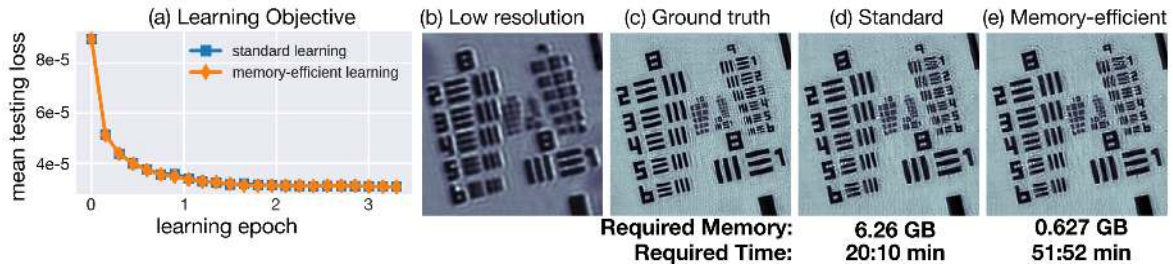

Figure 1. Super-resolution Microscopy: Comparison between (a) mean testing loss for standard and memory-efficient learning techniques. Visualization of (b) low-resolution, (c) ground truth reconstruction using all (89) measurements, and reconstruction using 8 measurements learned using (d) standard (with 6.26 GB and 20:10 min) and (e) memory-efficient learning (with 0.627 GB and 51:52 min).

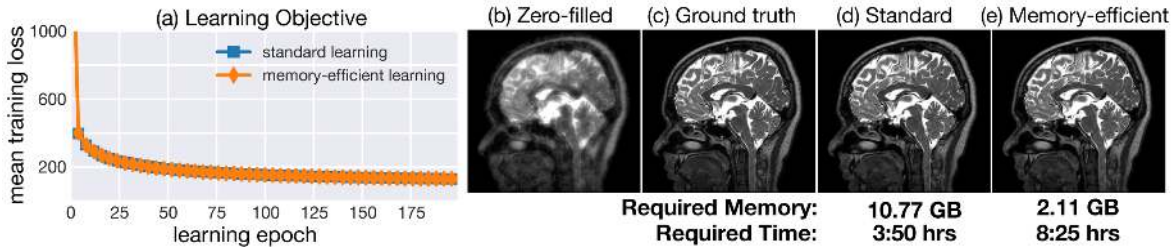

Figure 2. Multi-channel MRI: Comparison between (a) mean training loss for standard and memory-efficient learning techniques. Visualization of (b) zero-filled reconstruction, (c) ground truth reconstruction using fully sampled measurements, and PbN reconstruction learned using (d) standard (with 10.77 GB and 3:50 hours) and (e) memory-efficient learning (with 2.11 GB and 8:25 hours).

physiological dynamics. Multi-channel MRI is the standard of care in clinical systems and uses multiple receive coils distributed around the body to acquire measurements in parallel, thereby reducing the total number of required acquisition frames for decoding (18). By additionally modifying the measurement pattern to take advantage of image prior knowledge, *e.g.* through compressed sensing (19), it is possible to dramatically reduce scan times. As with experimental design, PbNs with learned deep image priors have demonstrated state-of-the-art performance for multi-channel MRI (20; 6), but are limited in network size and number of unrolled iterations due to memory required for training. Our memory-efficient learning reduces memory footprint at training time, thereby enabling learning for larger problems.

To evaluate our proposed memory-efficient learning, we reproduce the results in (6) for the "SD-ET-WD" PbN, which is equivalent to PGD (10 unrolled iterations) where the proximal update is replaced with a learned invertible residual convolutional neural network (RCNN) (21; 11; 9). We compare training with full backpropagation, requiring 10.77GB of memory and 3:50 hours, versus memory-efficient learning, requiring 2.11GB and 8:25 hours. We set $T = 6$ and do not use checkpointing. As Fig. 2 shows, the training loss is comparable across epochs, and inference results are similar on one image in the training set, with normalized root mean-squared error of 0.03 between conventional and memory-efficient learning.

6. Remarks

Discussion: Our proposed memory-efficient learning opens the door to applications that are not otherwise possible to train due to GPU memory constraints, without a large increase in training time. While we specialized the procedure to PGD networks, similar approaches can be taken to invert other PbNs with more complex subroutines such as solving linear systems of equations. However, sufficient conditions for invertibility must be met. This limitation is clear in the case of a gradient descent block with an evolving step size, as the Lipschitz constant may no longer satisfy Eq. 7. Furthermore, the convergent behavior of optimization to minima makes accurate reverse recalculation of unstored variables severely ill-posed and can cause numerical error accumulation. Checkpoints can be used to improve the accuracy of reverse recalculated variables, though most PbN are not deep enough for numerical convergence to occur.

Conclusion: In this communication, we presented a practical memory-efficient learning method for large-scale computational imaging problems without dramatically increasing training time. Using the concept of reversibility, we implemented reverse-mode differentiation with favorable spatial and temporal complexities. We demonstrated our method on two representative applications: SR optical microscopy and multi-channel MRI. We expect other computational imaging systems to nicely fall within our framework.

References

[1] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 399–406.

[2] J. Sun, H. Li, Z. Xu *et al.*, "Deep ADMM-Net for compressive sensing MRI," in *Advances in Neural Information Processing Systems*, 2016, pp. 10–18.

[3] M. Kellman, E. Bostan, N. Repina, and L. Waller, "Physics-based learned design: Optimized coded-illumination for quantitative phase imaging," *IEEE Transactions on Computational Imaging*, vol. 5, no. 3, pp. 344–353, 2019.

[4] M. Kellman, E. Bostan, M. Chen, and L. Waller, "Data-driven design for fourier ptychographic microscopy," in *Proceedings of the International Conference on Computational Photography*, 2019.

[5] V. Sitzmann, S. Diamond, Y. Peng, X. Dun, S. Boyd, W. Heidrich, F. Heide, and G. Wetzstein, "End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–13, Jul. 2018.

[6] H. K. Aggarwal, M. P. Mani, and M. Jacob, "Modl: Model-based deep learning architecture for inverse problems," *IEEE transactions on medical imaging*, vol. 38, no. 2, pp. 394–405, 2018.

[7] S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein, "Unrolled optimization with deep priors," *arXiv preprint arXiv:1705.08041*, 2017.

[8] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008.

[9] J. Behrmann, D. Duvenaud, and J.-H. Jacobsen, "Invertible residual networks," *arXiv preprint arXiv:1811.00995*, 2018.

[10] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in neural information processing systems*, 2018, pp. 6571–6583.

[11] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, "The reversible residual network: Backpropagation without storing activations," in *Advances in neural information processing systems*, 2017, pp. 2214–2224.

[12] B. Chang, L. Meng, E. Haber, L. Ruthotto, D. Begert, and E. Holtham, "Reversible architectures for arbitrarily deep residual neural networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[13] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[14] S. Banach, "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales," *Fund. math*, vol. 3, no. 1, 1922.

[15] G. Zheng, R. Horstmeyer, and C. Yang, "Wide-field, high-resolution Fourier ptychographic microscopy," *Nature Photonics*, vol. 7, no. 9, pp. 739–745, Jul. 2013.

[16] Z. F. Phillips, R. Eckert, and L. Waller, "Quasi-dome: A self-calibrated high-NA LED illuminator for Fourier ptychography," in *Imaging and Applied Optics 2017*. Optical Society of America, Jun. 2017.

[17] L. Tian, Z. Liu, L.-H. Yeh, M. Chen, J. Zhong, and L. Waller, "Computational illumination for high-speed in vitro Fourier ptychographic microscopy," *Optica*, vol. 2, no. 10, pp. 904–908, Oct. 2015.

[18] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger, "SENSE: sensitivity encoding for fast MRI," *Magn. Reson. Med.*, vol. 42, no. 5, pp. 952–962, Nov. 1999.

[19] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, Dec. 2007.

[20] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, "Learning a Variational Network for Reconstruction of Accelerated MRI Data," *Magnetic Resonance in Medicine*, vol. 79, no. 6, pp. 3055–3071, Nov. 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.