COMP 5704: Parallel Algorithms and Applications in Data Science

Fall 2021

Project on

Newton-ADMM: A Distributed GPU-Accelerated Optimizer for Non-Convex Problems

Chih-Hao Fang et al

Presented by

Kuntal Ghosh

Agendas

- Background
- Alternating Direction Method of Multipliers (ADMM) Framework
- Inexact Newton-CG Solver
- GPU-accelerated Newton-type Method
- Proposed Modification
- Discussion
- References
- Vote of Thanks

Background

Optimization : It is a technique of maximizing or minimizing a real function by methodically choosing input values from within an allowed set and computing the value of the function. For example, Gradient Decent or Stochastic Gradient Decent (SGD)

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \triangleq \sum_{i=1}^{n} f_i(\mathbf{x}) + g(\mathbf{x})$$

uOttawa

Background

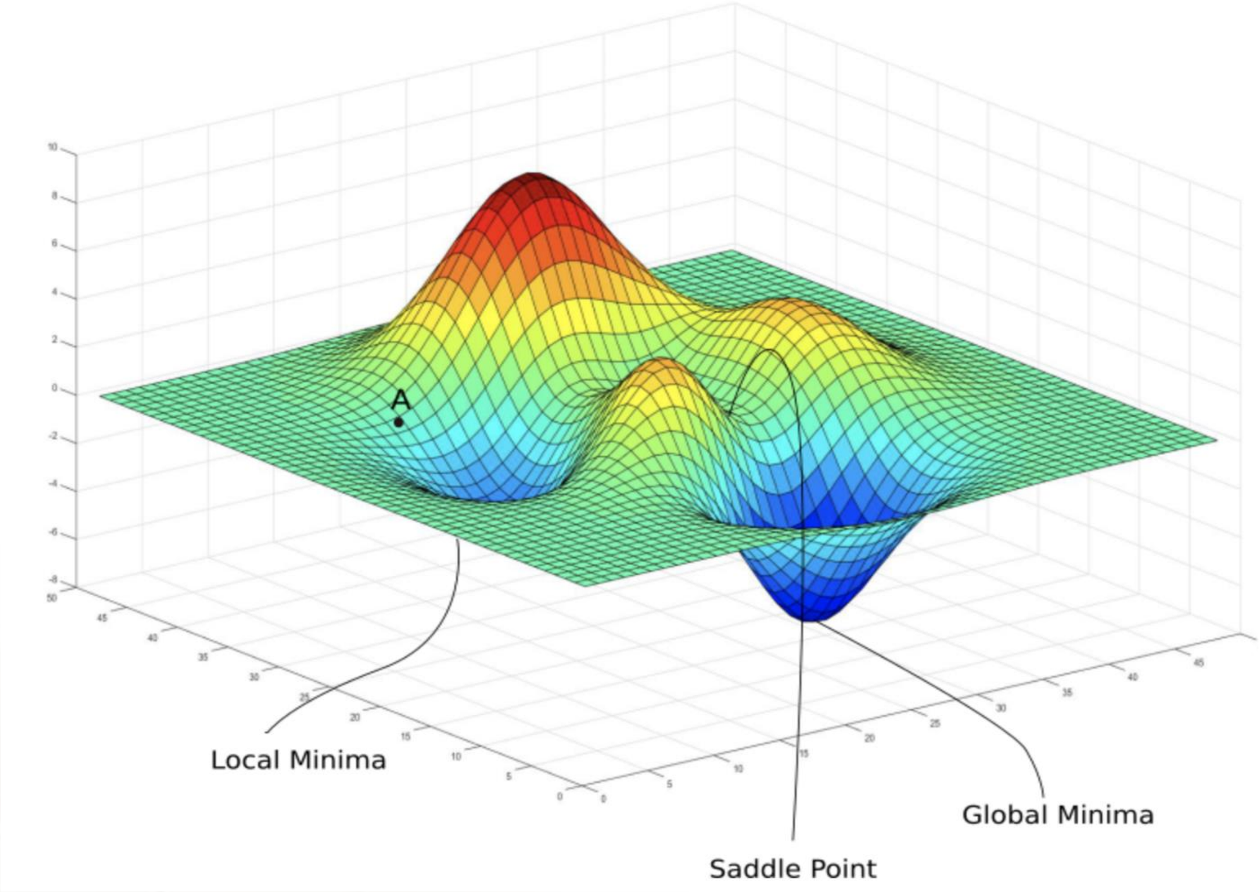

Fig 1 (10)

Background

Key challenges in optimization for machine learning problems.

- huge numbers of iterations having related communication costs in distributed environments.

- It is not easy to gather the whole training set at a single node and process the entire data serially due to the lack of proper resources.

- Privacy of the data

Background

Strategies of Existing Distributed Optimization Solvers

- Executing each operation in conventional solvers (e.g., SGD or (quasi) Newton) in a distributed environment.

- Executing local optimization methods which operate on their own data together having a coordinating method which synchronizes the models over iterations.

Alternating Direction Method of Multipliers (ADMM) Framework

It is a well-known technique in distributed optimization for solving consensus problems.

Let $\mathcal{N}$ denote the number of nodes (computing elements) in the distributed environment. Assume that the input dataset $\mathcal{D}$ is split among the $\mathcal{N}$ nodes such that :

$$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \ldots \cup \mathcal{D}_{\mathcal{N}}$$

$$\min \sum_{i=1}^{\mathcal{N}} \sum_{j \in \mathcal{D}_i} f_j(\mathbf{x}_i) + g(\mathbf{z})$$

$$\text{s.t.} \quad \mathbf{x}_i - \mathbf{z} = 0, \quad i = 1, \ldots, \mathcal{N}$$

uOttawa

Alternating Direction Method of Multipliers (ADMM) Framework

Algorithm 1: ADMM method (outer solver)

Input : $\mathbf{x}^{(0)}$ (initial iterate), $\mathcal{N}$ (no. of nodes)
Parameters: β, λ and $\theta < 1$

```
Initialize z^0 to 0
Initialize y_i^0 to 0 on all nodes.
foreach k = 0, 1, 2, ... do
    Perform Algorithm 2 with, x_i^k, y_i^k, and z^k on all
      nodes
    Collect all local x_i^{k+1}
    Evaluate z^{k+1} and y_i^{k+1}
    Distribute z^{k+1} and y_i^{k+1} to all nodes.
    Locally, on each node, compute spectral step sizes
      and penalty parameters
end
```

Alternating Direction Method of Multipliers (ADMM) Framework

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} f_i(\mathbf{x}_i) + \frac{\rho_i^k}{2}||\mathbf{z}^k - \mathbf{x}_i + \frac{\mathbf{y}_i^k}{\rho_i^k}||_2^2,$$

$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} g(\mathbf{z}) + \sum_{i=1}^{\mathcal{N}} \frac{\rho_i^k}{2}||\mathbf{z} - \mathbf{x}_i^{k+1} + \frac{\mathbf{y}_i^k}{\rho_i^k}||_2^2,$$

$$\mathbf{y}_i^{k+1} = \mathbf{y}_i^k + \rho_i^k(\mathbf{z}^{k+1} - \mathbf{x}_i^{k+1}).$$

Alternating Direction Method of Multipliers (ADMM) Framework

$$g(\mathbf{x}) = \lambda \|\mathbf{x}\|^2/2$$

$$\mathbf{z}^{k+1}(\lambda + \sum_{i=1}^{\mathcal{N}} \rho_i^k) = \sum_{i=1}^{\mathcal{N}} [\rho_i^k \mathbf{x}_i^{k+1} - \mathbf{y}_i^k]$$

Inexact Newton-CG Solver

Algorithm 2: Inexact Newton-type Method

```
Input      : x^(0)
Parameters: 0 < β, θ < 1
foreach k = 0, 1, 2, ... do
    Form g(x^(k)) and H(x^(k))
    if ||g(x^(k))|| < ε then
        STOP
    end
    Update x^(k+1)
end
```

Inexact Newton-CG Solver

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}_k$$

$$\left\| \mathrm{H}\left(\mathrm{x}^{(k)}\right) \mathrm{p}_k + \mathrm{g}\left(\mathrm{x}^{(k)}\right) \right\| \leq \theta \left\| \mathrm{g}\left(\mathrm{x}^{(k)}\right) \right\|$$

uOttawa

Inexact Newton-CG Solver

$$\mathbf{g}(\mathbf{x}) \triangleq \sum_{j \in \mathcal{D}} \nabla \hat{f}_j(\mathbf{x})$$

$$\mathbf{H}(\mathbf{x}) \triangleq \sum_{j \in \mathcal{D}} \nabla^2 \hat{f}_j(\mathbf{x})$$

GPU-accelerated Newton-type Method

- A Hessian-free Newton-type method to solve the ADMM subproblem
- That is to compute the Hessian-vector product **Hv** without explicitly forming the Hessian **H**.
- The Hessianvector product, Hv, can be efficiently computed using GEneral Matrix to Matrix Multiplication (GEMM) operations.
- Pytorch's Basic Linear Algebra Subprograms(BLAS) interface to the GPUs is used for GEMM.

Proposed Modification

- Inexact Newton-CG solver can fail if there is no positive definite Hessian

- Introduction of non-convex solver Newton-MR for non-convex problems generated from deep neural networks.

- Newton-MR's iterations can be written as follwos

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k, \quad \text{with} \quad \mathbf{p}_k = -\left[\nabla^2 f(\mathbf{x}_k)\right] \nabla f(\mathbf{x}_k)$$

uOttawa

Discussion

Experimental Setup and Data

- Hardware : two different hardware platforms were used by the authors: a) first one is Intel Xeon Platinum 8168 processors and 8 Tesla P100 GPU cards b) second one is is a CentOS 7 cluster with 15 nodes with 100 Gbps Infiniband interconnect - node has 96GB RAM, two 12-Core Intel Xeon Gold processors, and 3 Tesla P100 GPU cards.
- Data : HIGGS, MNIST, CIFAR-10, E18 data sets were used to compare Newton-ADMM with state-of-the-art first-order and second-order optimizers

Classes	Dataset	Train Size	Test Size	Dims
2	HIGGS	10,000,000	1,000,000	28
10	MNIST	60,000	10,000	784
10	CIFAR-10	50,000	10,000	3,072
20	E18	1,300,128	6,000	279,998

Discussion

Results

	S1					S2			
MNIST	**CPU Time**	**GPU Time**	**Comm. Time**	**Train. Obj.**	**MNIST**	**CPU Time**	**GPU Time**	**Comm. Time**	**Train. Obj.**
Newton-ADMM	3676.00	66.90	0.081	0.22	Newton-ADMM	1462.36	36.54	0.28	0.23
SGD	461.62	1034.63	18.97	0.24	SGD	229.16	530.92	23.84	0.23
CIFAR-10					**CIFAR-10**				
Newton-ADMM	11419.11	176.64	0.59	1.63	Newton-ADMM	5321.67	99.28	0.85	1.63
SGD	883.16	1280.91	43.76	1.67	SGD	443.71	663.99	54.28	1.65
HIGGS					**HIGGS**				
Newton-ADMM	23098.02	2159.53	0.08	0.64	Newton-ADMM	12356.39	1107.97	0.24	0.64
SGD	36791.37	125935.64	1739.15	0.65	SGD	18305.36	61597.04	1237.57	0.65
	S4					**S8**			
MNIST	**CPU Time**	**GPU Time**	**Comm. Time**	**Train. Obj.**	**MNIST**	**CPU Time**	**GPU Time**	**Comm. Time**	**Train. Obj.**
Newton-ADMM	787.38	26.29	0.39	0.24	Newton-ADMM	260.71	18.19	0.72	0.24
SGD	115.88	26751	39.31	0.26	SGD	61.63	138.30	28.82	0.28
CIFAR-10					**CIFAR-10**				
Newton-ADMM	2482.32	48.48	1.480	1.66	Newton-ADMM	1227.98	33.20	2.67	1.68
SGD	217.714	331.17	88.62	1.65	SGD	109.68	168.61	60.80	1.67
HIGGS					**HIGGS**				
Newton-ADMM	3784.05	443.10	0.22	0.64	Newton-ADMM	1581.05	215.86	0.12	0.64
SGD	9018.30	29904.36	1523.00	0.65	SGD	4629.18	16157.91	666.69	0.65

Discussion

Results

	W1					W2			
MNIST	**CPU Time**	**GPU Time**	**Comm. Time**	**Train. Obj.**	**MNIST**	**CPU Time**	**GPU Time**	**Comm. Time**	**Train. Obj.**
Newton-ADMM	322.15	20.20	0.25	0.06	Newton-ADMM	310.03	17.85	0.31	0.178
SGD	60.58	133.94	8.67	0.09	SGD	61.03	152.75	10.75	0.16
CIFAR-10					**CIFAR-10**				
Newton-ADMM	1268.43	33.80	0.63	1.29	Newton-ADMM	1223.65	33.12	0.941	1.59
SGD	106.06	173.09	15.87	1.40	SGD	110.52	169.17	22.99	1.55
HIGGS					**HIGGS**				
Newton-ADMM	1612.71	226.69	0.07	0.64	Newton-ADMM	1740.02	202.35	0.07	0.64
SGD	4484.31	16169.06	220.46	0.65	SGD	4539.78	16437.12	297.99	0.65
E18					**E18**				
Newton-ADMM	60644.96	907.57	15.917	0.006	**Newton-ADMM**	84793.51	1003.51	19.66	0.007
SGD	10534.14	8084.51	1723.16	0.03	**SGD**	11433.75	8101.65	2366.56	0.058
	W4					**W8**			
MNIST	**CPU Time**	**GPU Time**	**Comm. Time**	**Train. Obj.**	**MNIST**	**CPU Time**	**GPU Time**	**Comm. Time**	**Train. Obj.**
Newton-ADMM	326.67	18.64	0.44	0.19	Newton-ADMM	260.71	18.19	0.72	0.24
SGD	57.63	142.86	15.39	0.20	SGD	61.63	138.30	28.82	0.28
CIFAR-10					**CIFAR-10**				
Newton-ADMM	1251.88	33.17	1.86	1.65	Newton-ADMM	1227.98	33.20	2.67	1.68
SGD	110.59	171.48	34.53	1.63	SGD	109.68	168.61	60.80	1.67
HIGGS					**HIGGS**				
Newton-ADMM	1444.01	212.38	0.08	0.64	Newton-ADMM	1581.05	215.86	0.12	0.64
SGD	4574.42	16272.24	445.67	0.65	SGD	4629.18	16157.91	666.69	0.65
E18					**E18**				
Newton-ADMM	74356.35	1015.58	55.03	0.05	Newton-ADMM	79368.43	1003.86	94.43	0.09
SGD	9442.47	6195.97	6317.41	0.08	SGD	9558.39	5882.13	6611.72	0.10

Discussion

Q & A

References

[1] Fred Roosta, Yang Liu, Peng Xu, and Michael W Mahoney. “Newton-MR: Newton’s Method Without Smoothness or Convexity”. In: arXiv preprint arXiv:1810.00303 (2018).

[2] Albert S Berahas, Raghu Bollapragada, and Jorge Nocedal. “An Investigation of Newton-Sketch and Subsampled Newton Methods”. In: arXiv preprint arXiv:1705.06211 (2017).

[3] L. Angelani, R. Di Leonardo, G. Ruocco, A. Scala, and F. Sciortino. Saddles in the Energy Landscape Probed by Supercooled Liquids. Physical review letters, 85(25):5356, 2000.

[4] S ́ebastien Bubeck et al. “Convex optimization: Algorithms and complexity”. In: Foundations and Trends® in Machine Learning 8.3-4 (2015), pp. 231–357.

[5] Y. Arjevani, O. Shamir, and R. Shi. Oracle Complexity of Second-Order Methods for Smooth Convex Optimization. Mathematical Programming, pages 1{34, 2017.

[6] Jianmin Chen, Xinghao Pan, Rajat Monga, Samy Bengio, and Rafal Jozefowicz. “Revisiting distributed synchronous SGD”. In: arXiv preprint arXiv:1604.00981 (2016).

[7]Y. Bengio et al. Learning deep architectures for AI. Foundations and trends R in Machine Learning, 2(1):1{127, 2009.

[8] S. Bellavia, C. Cartis, N. I. M. Gould, B. Morini, and Ph. L. Toint. Convergence of a regularized Euclidean residual algorithm for nonlinear least-squares. SIAM Journal on Numerical Analysis, 48(1):1{29, 2010.

[9] Groueix, T., Fisher, M., Kim, V., Russell, B., Aubry, M.: Atlasnet: A papier-m^ache approach to learning 3d surface generation. In: CVPR 2018 (2018)

References

[10] Rixon Crane and Fred Roosta. "DINGO: Distributed Newton-Type Method for Gradient-Norm Optimization". In: Proceedings of the Advances in Neural Information Processing Systems. Accepted. 2019.

[11] Hadi Daneshmand, Aurelien Lucchi, and Thomas Hofmann. "DynaNewton-Accelerating Newton's Method for Machine Learning". In: arXiv preprint arXiv:1605.06561 (2016).

[12] D. Calvetti, B. Lewis, and L. Reichel. L-curve for the MINRES method. In Advanced Signal Processing Algorithms, Architectures, and Implementations X, volume 4116, pages 385{396. International Society for Optics and Photonics, 2000.

[13] Priya Goyal, Piotr Doll ́ar, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. "Accurate, large minibatch SGD: training imagenet in 1 hour". In: arXiv preprint arXiv:1706.02677 (2017).

[14] K. v. d. Doel and U. Ascher. Adaptive and stochastic algorithms for EIT and DC resistivity problems with piecewise constant solutions and many measurements. SIAM J. Scient. Comput., 34:DOI: 10.1137/110826692 2012

[15]Sashank J Reddi, Jakub Koneˇcn`y, Peter Richt ́arik, Barnab ́as P ́ocz ́os, and Alex Smola. "AIDE: Fast and communication efficient distributed optimization". In: arXiv preprint arXiv:1608.06879 (2016).

[16] . B. Kylasa, F. Roosta-Khorasani, M. W. Mahoney, and A. Grama. GPU Accelerated Sub-Sampled Newton's Method. arXiv preprint arXiv:1802.09113. Accepted for publication in the Proceedings of SIAM SDM 2019.

[17] Peng Xu, Farbod Roosta-Khorasani, and Michael W. Mahoney. "Second-Order Optimization for Non-Convex Machine Learning: An Empirical Study". In: arXiv preprint arXiv:1708.07827 (2017).

[18] H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak- Lojasiewicz condition. Joint European Conference on Machine Learning and Knowledge Discovery in Databases,pages 795{811, 2016.

uOttawa

Thanks a googol!