Deep Learning with Nontrivial Constraints

Buyun Liang*, Ryan de Vera*, Tim Mitchell†, and Ju Sun*

Abstract. This tutorial focuses on practical ways to handle constraints in deep learning and its applications. We will start with constraints that can be absorbed into deep neural networks, then move to simple constraints that allow projected-gradient style algorithms. For nontrivial constraints, we will discuss standard numerical methods such as penalty methods and augmented Lagrangian methods. Our tutorial will culminate with the introduction of `NCVX`, a general-purpose optimization package we have built to solve generic constrained deep learning problems. We will draw concrete examples from various scientific and engineering domains such as computer vision, structure design, physics-aware machine learning, and imbalanced learning, to help the audience to understand and apply these practical numerical methods.

1. Basic Information. Imposing explicit constraints is relatively new but increasingly pressing in deep learning, stimulated by, e.g., trustworthy AI that performs robust optimization over complicated perturbation sets [11, 19, 10, 12, 5, 6, 17] and scientific and engineering applications that need to respect physical laws and constraints [3, 2, 4, 13, 7, 9, 18, 20]. However, it can be hard to reliably solve constrained deep learning problems without optimization expertise. Existing deep learning frameworks, such as TensorFlow [1] and PyTorch [21], do not admit constraints. General-purpose optimization packages can handle constraints but do not perform auto-differentiation and have trouble dealing with nonsmoothness [23, 14, 8, 22]. In this tutorial, we will introduce various applications of constrained deep learning in science and engineering, and also practical ways (e.g., projected gradient methods [6, 5], penalty methods, augmented Lagrangian methods [7, 9, 18, 20]) to solve these types of problems. In particular, we will highlight a user-friendly optimization package `NCVX` [15, 16] that we have built specifically for solving constrained deep learning painlessly, and discuss practical tricks to speed up its convergence in applications [17].

2. Target Audience. This tutorial targets applied AI practitioners and researchers, with or without a technical background in constrained optimization. The audience is expected to be familiar with the basic concepts in machine and deep learning. It is especially beneficial for audiences 1) who are new to constrained deep learning and want to learn the basics of this field quickly; 2) who want to learn the state-of-the-art works in constrained deep learning; 3) who encounter constrained deep learning problems in their research and look for quick and reliable numerical solvers. This tutorial will be delivered at the fresh graduate level and will be easily accessible to both industrial and academic AI researchers and practitioners.

3. Tutorial Structure. We will introduce both scientific and engineering applications leading to constrained deep learning problems with nontrivial constraints, and practical numerical methods to solve them. We will start with the background and motivation of deep learning with nontrivial constraints, followed by the current challenges about solving this type of problems. Next, we will introduce the recent works in constrained deep learning, which is necessary

*Computer Science & Engineering, University of Minnesota. {liang664,dever120,jusun}@umn.edu
†Department of Computer Science, Queens College, City University of New York. tmitchell@qc.cuny.edu

in robust vision recognition and AI for science, but is not easy to solve. After that, we will focus on a general-purpose software package targeted at constrained deep learning. Last, we will discuss some open problems and the future directions to go.

1. Background and Motivation (20 min)
    (a) Motivating examples: Robustness in vision recognition, AI for science (10 min)
    (b) Challenges: Reliably solving them requires optimization expertise (10 min)
2. Concrete Examples of Constrained Deep Learning & Tailored Numerical Methods for Solving Them (40 min)
    (a) Robustness in vision recognition (10 min)
    (b) Knowledge-aware machine learning (10 min)
    (c) Neural structural optimization (10 min)
    (d) Orthogonal recurrent neural networks (10 min)
3. Break (5 min)
4. `NCVX`: A General-Purpose Software Package for Constrained Deep Learning (50 min)
    (a) Algorithms of `NCVX` (15 min)
    (b) Constrained DL examples in `NCVX` (15 min)
    (c) Practical tricks to speed up convergence (20 min)
5. Open Problems and Frontiers (10 min)
    (a) Challenges (5 min)
    (b) Future work (5 min)

4. Tutor's Bios.

Buyun Liang. is a MS student of computer science at UMN, where he worked as a graduate researcher at the GLOVEX group, led by Prof. Ju Sun. Previously he obtained his bachelor's degree in physics at Nanjing University, and also a master's degree in materials science at UMN, where his research focus is about Monte-Carlo and molecular dynamics simulation. He is the lead author of `NCVX`, the general-purpose software package targeted at constrained deep learning. He also focuses on customizing `NCVX` for different practical problems, such as robustness for vision recognition and AI for science. See https://buyunliang.org for more information.

Ryan Devera. is a first-year PhD student in Computer Science & Engineering, UMN, working with Prof. Ju Sun on constrained deep learning and AI for science and engineering at large. Before this, he worked for eight years as a senior data scientist, project manager, and technical mentor in various start-up companies. He holds a master degree in applied mathematics and bachelor degrees in mathematics and physics.

Prof. Tim Mitchell. is an assistant professor of computer science at Queens College/CUNY. His research interests span the areas of optimization, numerical linear algebra, and scientific computing, with one focus being computing and optimizing robustness properties of linear dynamical systems. He is also interested in nonsmooth constrained optimization, machine learning, and model-order reduction. He was a postdoc at the Max Planck Institute in Magdeburg, Germany and the Courant Institute at NYU, which is where he did his PhD, and he previously worked at IBM Thomas J. Watson Research Center in Hawthorne, New York. For more info, see http://www.timmitchell.com.

Prof. Ju Sun. is an assistant professor at the Department of Computer Science Engineering, the University of Minnesota at Twin Cities. His research interests span computer vision, machine learning, numerical optimization, data science, computational imaging, and healthcare. His recent efforts are focused on the foundation and computation for deep learning and applying deep learning to tackle challenging science, engineering, and medical problems. Before this, he worked as a postdoc scholar at Stanford University (2016-2019), obtained his Ph.D. degree from Electrical Engineering of Columbia University in 2016 (2011-2016), and B.Eng. in Computer Engineering (with a minor in Mathematics) from the National University of Singapore in 2008 (2004-2008). He won the best student paper award from SPARS'15, honorable mention of doctoral thesis for the New World Mathematics Awards (NWMA) 2017, and AAAI New Faculty Highlight Programs 2021.

5. History. This is the first time we will present this tutorial. We plan to submit similar tutorial proposals to other top machine learning, computer vision, and relevant scientific and engineering conferences, with the applications tailored more to their domains.

REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*, arXiv preprint arXiv:1603.04467, (2016).

[2] A. Chandrasekhar, S. Sridhara, and K. Suresh, *Auto: a framework for automatic differentiation in topology optimization*, Structural and Multidisciplinary Optimization, 64 (2021), pp. 4355–4365.

[3] A. Chandrasekhar and K. Suresh, *Tounn: topology optimization using neural networks*, Structural and Multidisciplinary Optimization, 63 (2021), pp. 1135–1149.

[4] P. W. Christensen and A. Klarbring, *An introduction to structural optimization*, vol. 153, Springer Science & Business Media, 2008.

[5] F. Croce and M. Hein, *Minimally distorted adversarial examples with a fast adaptive boundary attack*, in International Conference on Machine Learning, PMLR, 2020, pp. 2196–2205.

[6] F. Croce and M. Hein, *Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks*, in International conference on machine learning, PMLR, 2020, pp. 2206–2216.

[7] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, *Scientific machine learning through physics-informed neural networks: Where we are and what's next*, arXiv preprint arXiv:2201.05624, (2022).

[8] R. R. Curtin, M. Edel, R. G. Prabhu, S. Basak, Z. Lou, and C. Sanderson, *The ensmallen library for flexible numerical optimization.*, J. Mach. Learn. Res., 22 (2021), pp. 166–1.

[9] A. Dener, M. A. Miller, R. M. Churchill, T. Munson, and C.-S. Chang, *Training neural networks under physical constraints using a stochastic augmented lagrangian approach*, arXiv preprint arXiv:2009.07330, (2020).

[10] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, *Exploring the landscape of spatial robustness*, in International conference on machine learning, PMLR, 2019, pp. 1802–1811.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, arXiv preprint arXiv:1412.6572, (2014).

[12] D. Hendrycks and T. Dietterich, *Benchmarking neural network robustness to common corruptions and perturbations*, arXiv preprint arXiv:1903.12261, (2019).

[13] S. Hoyer, J. Sohl-Dickstein, and S. Greydanus, *Neural reparameterization improves structural optimization*, arXiv preprint arXiv:1909.04240, (2019).

[14] S. Laue, M. Mitterreiter, and J. Giesen, *Geno–generic optimization for classical machine learning*, Advances in Neural Information Processing Systems, 32 (2019).

[15] B. Liang, T. Mitchell, and J. Sun, *Ncvx: A user-friendly and scalable package for nonconvex opti-*

mization in machine learning, arXiv preprint arXiv:2111.13984, (2021).
[16] B. Liang, T. Mitchell, and J. Sun, *Ncvx: A general-purpose optimization solver for constrained machine and deep learning*, arXiv preprint arXiv:2210.00973, (2022).
[17] H. Liang, B. Liang, Y. Cui, T. Mitchell, and J. Sun, *Optimization for robustness evaluation beyond ell_p metrics*, arXiv preprint arXiv:2210.00621, (2022).
[18] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, *Deepxde: A deep learning library for solving differential equations*, SIAM Review, 63 (2021), pp. 208–228.
[19] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards deep learning models resistant to adversarial attacks*, arXiv preprint arXiv:1706.06083, (2017).
[20] L. McClenny and U. Braga-Neto, *Self-adaptive physics-informed neural networks using a soft attention mechanism*, arXiv preprint arXiv:2009.04544, (2020).
[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., *Pytorch: An imperative style, high-performance deep learning library*, Advances in neural information processing systems, 32 (2019).
[22] G. Pillo and M. Roma, *Large-scale nonlinear optimization*, vol. 83, Springer Science & Business Media, 2006.
[23] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical programming, 106 (2006), pp. 25–57.