Machine Learning and Neural Networks for Field Theory

Sam Foreman*[1], Xiao-Yong Jin[1], and James C. Osborn[1]

[1]Argonne National Laboratory

(DRAFT) August 31, 2020

Here we outline some of the research in applying ML methods to field theory with an emphasis on the MCMC sampling problem in lattice QCD where we have recently been working.

1 Machine learning for Markov chain Monte Carlo

Perturbative acceleration [1–3].

The ability to efficiently sample from high-dimensional distributions remains a widely-pursued goal across scientific disciplines, with some noteable examples including particle physics, molecular dynamics, protein folding, population genetics, neuroscience, epidemiology, economics, ecology, climate science, and astrophysics & cosmology [4]. Recently, there has been a growing interest in developing probabilistic models that are parameterized by neural networks, and while much progress has been made in this direction [5–10], mitigating the *critical slowing down* (CSD) effect for lattice QCD remains a long-term goal of the community.

Since all lattice QCD simulations are performed at finite lattice spacing a, an extrapolation to the continuum limit is required in order to accurately compute physical quantities of interest. More reliable extrapolations can be done by simulating the theory at increasingly smaller lattice spacings while keeping the physics constant. Unfortunately, this causes the correlation times of these quantities to diverge, indicating that the continuum limit is a critical point of the theory.

Markov Chain Monte Carlo (MCMC) algorithms are known to encounter difficulties when simulating theories near a critical point, an issue known as the *critical slowing down* of the algorithm [11]. This effect can easily be seen in the topological charge $\mathcal{Q} \in \mathbb{Z}$, whose auto-correlation time increases dramatically with smaller lattice spacing as configurations tend to get 'stuck' in distinct topological sectors, preventing an efficient exploration of the phase space. As a result, developing new sampling techniques that are able to offer improvements in efficiency through a reduction of statistical autocorrelations are highly desired. In this LOI, we describe some recent work in this area, and provide suggestions for possible future directions. Generally speaking, MCMC methods are a class of algorithms that use Markov Chains to sample from a *target distribution* $p(x)$ that is often too complicated to sample from directly. Currently, the Hamiltonian Monte Carlo (HMC) algorithm is the most widely used technique for generating gauge configurations in lattice gauge theory and lattice QCD. We include below a brief overview of the approach, but refer the interested reader to [12, 13] for more details and limitations.

We begin by introducing an auxiliary momentum variable v (normally distributed, independent of the position x) in order to lift the target distribution onto a joint probability distribution $p(x, v)$ in phase space. The Markov Chain is then obtained by simulating a physical system governed by a Hamiltonian comprised of kinetic and potential energy functions, i.e. $\mathcal{H}(x, v) = U(x) + T(v)$. In particular, HMC operates by sampling from the canonical distribution $p(x, v) = \exp(\mathcal{H}(x, v)) = p(x)p(v)$ by solving the equations of motion ($\dot{x}_i = \frac{\partial \mathcal{H}}{\partial v_i}, \dot{v}_i = -\frac{\partial \mathcal{H}}{\partial x_i}$) for a fixed period of time using a volume-preserving integrator. In practice the integration is done in discrete steps introducing some numerical error. This then requires a Metropolis accept/reject step to correct for the error.

As the lattice spacing decreases, sectors of different topology become separated by large potential barriers, and simply moving along trajectories from the standard EOM become inefficient at moving between

*foremans@anl.gov

topological sectors. There are several ways to modify the EOM to help with this. One option is to incorporate extra functions into the EOM that can deform the trajectories so that tunneling is more efficient. These extra functions can be parameterized as neural networks that can be trained during sampling.

In the L2HMC algorithm [5], the authors propose a learned inference architecture that exhibits many highly-desirable properties, including: fast mixing (i.e. the ability to quickly produce uncorrelated samples); fast burn-in (i.e. rapid convergence to the target distribution); and the ability to mix across energy levels and between modes.

This is done by introducing six new functions, $S_\ell(\theta), T_\ell(\theta), Q_\ell(\theta)$ for $\ell = x, v$ into the leapfrog equations, each of which are parameterized by weights θ in a deep neural network. These weights can then be trained 'on the fly' by minimizing a suitably chosen loss function. While the proposed modifications to the integrator have the potential for violating the symplectic requirement of HMC, careful bookkeeping of the separate Jacobian factors in the acceptance criteria ensures that the algorithm remains statistically exact, and will (asymptotically) converge to the correct target distribution. The details of the augmented leapfrog equations can be found in the original work [5], but it can be shown [11] that the proposed modifications may allow for acceleration in low-density zones to facilitate mixing between modes. and better conditioning of the energy landscape (e.g. by learning a diagonal inertia tensor).

In [5], the authors propose a loss function designed to reduce the mixing time. This is done by *minimizing* the lag-one autocorrelation, or equivalently, by *maximizing* the expected squared jumped distance (ESJD) [14]. In $\mathbb{R}^n$, this can be (roughly) understood as maximizing the expected value of the Euclidean distance $\mathcal{L}(\theta) \simeq \mathbb{E}\left[A(\xi'|\xi) \cdot \|x' - x\|_2^2\right]$, between successive states in the chain.

While this choice of loss function works well enough in $\mathbb{R}^n$, it is not immediately obvious how it should be defined when working in the gauge groups of lattice QCD (e.g. $U(1), SU(2), SU(3)$, etc). Additionally, the generic network structure proposed in [5] is composed entirely of dense (fully-connected) layers, which are not particularly well-suited for data with an inherent geometry, or that exist on a non-Riemannian manifold. We have made progress towards dealing with both of these limitations in the case of a two-dimensional $U(1)$ gauge model defined on a square lattice with periodic boundary conditions [11, 15], but there remains much to be done. An open source implementation of the L2HMC algorithm along with its modifications for dealing with lattice gauge models is freely available [16].

2 Machine learning for sign problem

Path optimization for sign problem [17–21].

3 Explainable machine learning for studying physical systems

Explainable machine learning in lattice field theory, constructing observable characterizing the symmetric phase of the 2+1 D Yukawa model [22].

4 Machine learning for improving and discovering new algorithms

Field theory researchers have studied neural networks for multigrid algorithms in gauge fields [23]. Both algorithms in neural networks and multigrid in gauge fields have improved significantly since then, along with the increases in computational power, neural networks started to show promises in algorithmic designs. A type of shallow neural networks, the Boltzmann machine, originally designed to mimic physical systems, give researchers different cluster Monte Carlo algorithms after training [24]. More researches in deep neural networks and their application in algorithmic designs for lattice field theories will accelerate theoretical computations.

5 Machine learning for theoretical advancement

Tensor renormalization group replaced by neural networks [25].

References

[1] S. Duane, R. Kenway, B. J. Pendleton, and D. Roweth, "Acceleration of Gauge Field Dynamics", Phys. Lett. **B176**, 143 (1986).

[2] S. Duane and B. J. Pendleton, "Gauge Invariant Fourier Acceleration", Phys. Lett. **B206**, 101 (1988).

[3] G. Cossu, P. Boyle, N. Christ, C. Jung, A. Jüttner, and F. Sanfilippo, "Testing algorithms for critical slowing down", EPJ Web Conf. **175**, 02008 (2018).

[4] K. Cranmer, J. Brehmer, and G. Louppe, "The frontier of simulation-based inference", Proceedings of the National Academy of Sciences (2020).

[5] D. Levy, M. D. Hoffman, and J. Sohl-Dickstein, "Generalizing Hamiltonian Monte Carlo with Neural Networks", arXiv e-prints, arXiv:1711.09268, arXiv:1711.09268 (2017).

[6] M. Albergo, G. Kanwar, and P. Shanahan, "Flow-based generative models for markov chain monte carlo in lattice field theory", Physical Review D **100**, 034515 (2019).

[7] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, "Equivariant flow-based sampling for lattice gauge theory", arXiv preprint arXiv:2003.06413 (2020).

[8] D. J. Rezende, S. Racanière, I. Higgins, and P. Toth, "Equivariant hamiltonian flows", arXiv preprint arXiv:1909.13739 (2019).

[9] P. Toth, D. J. Rezende, A. Jaegle, S. Racanière, A. Botev, and I. Higgins, "Hamiltonian generative networks", arXiv preprint arXiv:1909.13789 (2019).

[10] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan, "Sampling using $su(n)$ gauge equivariant flows", arXiv preprint arXiv:2008.05456 (2020).

[11] S. A. Foreman, *Learning better physics: a machine learning approach to lattice gauge theory.*

[12] M. Betancourt, "A conceptual introduction to hamiltonian monte carlo", arXiv preprint arXiv:1701.02434 (2017).

[13] R. M. Neal et al., "Mcmc using hamiltonian dynamics", Handbook of markov chain monte carlo **2**, 2 (2011).

[14] C. Pasarica and A. Gelman, "Adaptively scaling the metropolis algorithm using expected squared jumped distance", Statistica Sinica, 343 (2010).

[15] S. A. Foreman, *Machine learning in lattice QCD*, 2020.

[16] *L2hmc-qcd*, `https://www.github.com/saforem2/l2hmc-qcd`.

[17] Y. Mori, K. Kashiwa, and A. Ohnishi, "Toward solving the sign problem with path optimization method", Phys. Rev. D **96**, 111501 (2017).

[18] A. Ohnishi, Y. Mori, and K. Kashiwa, "Path optimization method for the sign problem", EPJ Web Conf. **175**, 07043 (2018).

[19] A. Ohnishi, Y. Mori, and K. Kashiwa, "Path optimization method with use of neural network for the sign problem in field theories", PoS **LATTICE2018**, 023 (2018).

[20] A. Alexandru, P. F. Bedaque, H. Lamm, S. Lawrence, and N. C. Warrington, "Fermions at Finite Density in 2+1 Dimensions with Sign-Optimized Manifolds", Phys. Rev. Lett. **121**, 191602 (2018).

[21] K. Kashiwa, Y. Mori, and A. Ohnishi, "Application of the path optimization method to the sign problem in an effective model of QCD with a repulsive vector-type interaction", Phys. Rev. D **99**, 114005 (2019).

[22] S. Blücher, L. Kades, J. M. Pawlowski, N. Strodthoff, and J. M. Urban, "Towards novel insights in lattice field theory with explainable machine learning", Phys. Rev. D **101**, 094507 (2020).

[23] M. Bäker, G. Mack, and M. Speh, "Multigrid meets neural nets", Nuclear Physics B - Proceedings Supplements **30**, Proceedings of the International Symposium on, 269 (1993).

[24] L. Wang, "Exploring cluster monte carlo updates with boltzmann machines", Phys. Rev. E **96**, 051301 (2017).

[25] S.-H. Li and L. Wang, "Neural network renormalization group", Phys. Rev. Lett. **121**, 260601 (2018).