Machine learning action parameters for lattice QCD

Emmy Trewartha
Hybrid Monte-Carlo
Critical Slowing down

Multiscale Lattice Generation

From Endres et al, PRD 92 no. 11 114516 (2015)
Multiscale Lattice Generation - Parameter Matching

From Endres et al, PRD 94 no. 11 114502 (2016)
Can neural networks help?
Previous work: NNs able to learn SU(2) deconfinement transition using Polyakov loop

SJ Wetzel and M Scherzer, PRB96 no. 18 184410 (2017)
Neural Networks

Output of neuron \( Y = f(w_1 \cdot X_1 + w_2 \cdot X_2 + b) \)
Training set:

$12^3 \times 36$ SU(2) ensembles of 1000 configurations each

Two grids in $\beta$, $m$ space:

$\beta \in \{1.785, 1.835, 1.885, 1.935, 1.985\}$ and $m \in \{-0.7, -0.8, -0.9, -1.0\}$, excluding the pair $\{\beta, m\} = \{1.985, -1.0\}$

$\beta \in \{1.76, 1.81, 1.86, 1.91\}$ and $m \in \{-0.75, -0.85, -0.95, -1.05\}$, excluding the pair $\{\beta, m\} = \{1.91, -1.05\}$

850 randomly selected configurations used for training, 150 for validation
First Attempt: Configurations as input directly
Measure independence of configs using autocorrelation:

$$\rho(\tau) = \sum_{\tau'} \langle (O(\tau) - \langle O(\tau') \rangle)(O(\tau' + \tau) - \langle O(\tau' + \tau) \rangle) \rangle$$

At large $\tau$ behaves as:

$$\frac{\rho(\tau)}{\rho(0)} \approx \exp\left[-\frac{\tau}{\tau_{\text{exp}}} \right]$$

Then define

$$\tau_{\text{int}} = \frac{1}{2} + \lim_{\tau_{\text{max}} \to \infty} \frac{1}{\rho_0} \sum_{\tau=0}^{\tau_{\text{max}}} \rho(\tau)$$
Define:

\[ \rho(\tau) = \left[ P\alpha \left( c^\alpha(\tau) \right) + P\beta \left( c^\beta(\tau) \right) \right] - 1 \]

\[ \tau_{int} = \frac{1}{2} + \lim_{\tau_{max} \to \infty} \frac{1}{\rho_0} \sum_{\tau=0}^{\tau_{max}} \rho(\tau) \]

Generate ten independent streams of 10,000 trajectories denoted F1, \ldots, F10, saved every trajectory, generated with the same values of \( \beta = 1.76 \) and \( m_0 = -0.75 \).
Autocorrelation from classifier network
Interpretability - custom network design?

Sample Configuration → Localization Network → Averaging Layer → Prediction Network

SJ Wetzel and M Scherzer, PRB96 no. 18 184410 (2017)
Interpretability - interpretable mid-layers?
Interpretability - Pixelwise Relevance?

Interpretability - Pixelwise Relevance?
Sitewise Classifier Derivatives
Incorporating Symmetries in Network Design
Incorporating Symmetries in Network Design
Second attempt: Incorporate symmetries directly

\[ W_{j \times k, l \times m}(R) = \sum_{|r|=R} \sum_{\ell \in \mathcal{O}(j \times k)} \sum_{\ell' \in \mathcal{O}(l \times m)} \sum_x W_{\ell}(x) W_{\ell'}(x + r) \]
Symmetrized Loops

Correlated Products
Summary

Symmetry respecting neural networks are able to solve the lattice parameter regression problem well.

Fully connected networks reveal an unknown feature of longer correlation length than any observable studied.

Neural networks are able to learn non-trivial features of lattice gauge field theories.