

Autoregressive networks based on $d - 1$ dimensional convolutions for lattice field theory simulations

February 21, 2023

1 The autoregressive relation and scalar lattice field theory

The systems of interest consist of a box lattice of length L in d dimensions where every position is labeled using a d -dimensional vector $\mathbf{x} \in [1, L]^d$. The system state/configuration is described using scalar values at every position $\phi(\mathbf{x})$. The configurations obey the Boltzmann distribution:

$$p(\{\phi(\mathbf{x})\}_{\mathbf{x} \in [1, L]^d}) = e^{-S[\phi]}/Z \quad (1)$$

where the *action* $S[\phi]$ is a functional of the field values $\phi(\mathbf{x})$. The d -dimensional positions \mathbf{x} maybe replaced with a 1 dimensional ordering:

$$k = \left(\sum_{i=1}^d (x_i - 1)L^{i-1} \right) + 1 \quad (2)$$

where x_i are the components of \mathbf{x} and $k \in [1, N = L^d]$. Based on this ordering, we can write down the probability distribution in 1 as a product of conditional distributions at every position:

$$\begin{aligned} p(\{\phi_k\}) &= p(\phi_1, \phi_2 \dots \phi_N) = p(\phi_1)p(\phi_2|\phi_1) \dots p(\phi_N|\phi_{N-1} \dots \phi_2, \phi_1) \\ &= \prod_{k \in [1, N]} p(\phi_k|\phi_{<k}) \end{aligned} \quad (3)$$

This is the chain rule of conditional probabilities based on Bayes theorem or *autoregressive relation*. This mathematical relation is the basis of image and audio generation algorithms in deep learning such as MADE[4] and PixelCNN[7]. Our system of interest is the scalar lattice field theory whose action is given by:

$$S[\phi] = \sum_{\mathbf{x} \in [1, L]^d} \left[\phi(\mathbf{x}) \sum_{\mathbf{y}} \square(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) + m^2 \phi(\mathbf{x})^2 + \lambda \phi(\mathbf{x})^4 \right] \quad (4)$$

where a , m , λ are the lattice spacing, mass and coupling respectively. Assuming open boundary conditions, we can expand the d'Alembertian term in the RHS as:

$$\sum_{\mathbf{x} \in [1, L]^d} \phi(\mathbf{x}) \sum_{\mathbf{y}} \square(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) = \sum_{\mu=1}^d \sum_{x_\nu=\mu \in [2, L-1], x_\nu \neq [1, L]} 2\phi(\mathbf{x})^2 - \phi(\mathbf{x})\phi(\mathbf{x}-\hat{\mu}) - \phi(\mathbf{x})\phi(\mathbf{x}+\hat{\mu})$$

and $\phi(\mathbf{x})$ can take any real value. Note that the action S depends only on nearest neighbour product/interaction terms like $\phi(\mathbf{x})\phi(\mathbf{x}-\hat{\mu})$ besides powers of $\phi(\mathbf{x})$ alone.

2 Smaller dependency sets of conditional distributions due to nearest neighbour interactions

Examining the k th conditional probability $p(\phi_k | \phi_{<k})$ in 3, its distribution in general depends on $k-1$ values in $\phi_{<k} = \{\phi_{k-1}, \dots, \phi_1\}$. This means the complexity of these distributions can explode if the number of lattice points N is large, which is typically the case of interest. That's the reason deep neural networks have been utilized to model them for image/audio generation. However for systems with nearest neighbour interactions, the *dependency set* is significantly smaller (from my master's thesis [6]). It's easier to show this (without loss of generality) for the nearest neighbour Ising model whose action is given by:

$$S[\phi] = -\beta J \sum_{\mu=1}^d \sum_{x_\nu=\mu \in [2, L], x_\nu \neq \mu \in [1, L]} \phi(\mathbf{x}-\hat{\mu})\phi(\mathbf{x}) \quad (5)$$

where $\phi(\mathbf{x})$ takes values ± 1 . Restating the autoregressive relation for the Ising model¹:

$$\prod_{k=1}^N p(\phi_k | \phi_{<k}) = p(\phi) = \exp \left(-\beta J \sum_{\mu=1}^d \sum_k \phi_k \phi_{k-\hat{\mu}} \right) / Z \quad (6)$$

In order to determine the dependency set, let's start backwards from with the distribution for the N th spin $p(\phi_N | \phi_{<N})$ where all other $N-1$ spins are known. We can cluster together every other term in the above equation in the form an unconditional probability using Bayes theorem:

$$\prod_{k=1}^{N-1} p(\phi_k | \phi_{<k}) = p(\phi_{<N})$$

so that

$$p(\phi_N | \phi_{<N}) p(\phi_{<N}) = p(\phi)$$

¹ $k - \hat{\mu}$ should be understood as the lattice position $\mathbf{x} - \hat{\mu}$ where \mathbf{x} maps to k according to the given ordering

and the unconditional probability $\log p(\phi_{<N})$ can be written as:

$$p(\phi_{<N}) = \sum_{\phi_N} p(\phi) = \sum_{\phi_N} \exp \left(-\beta J \sum_{\mu=1}^d \sum_k \phi_k \phi_{k-\hat{\mu}} \right) / Z$$

which leads to

$$p(\phi_N | \phi_{<N}) = \frac{p(\phi)}{\sum_{\phi_N} p(\phi)} = \frac{\exp \left(-\beta J \sum_{\mu=1}^d \sum_k \phi_k \phi_{k-\hat{\mu}} \right)}{\sum_{\phi_N} \exp \left(-\beta J \sum_{\mu=1}^d \sum_k \phi_k \phi_{k-\hat{\mu}} \right)}$$

Here, every term cancels between the numerator and denominator except the ones that containing ϕ_N , since addition within exponentials beautifully factors outside:

$$p(\phi_N | \phi_{<N}) = \frac{\exp \left(-\beta J \phi_N \sum_{\mu=1}^d \phi_{k-\hat{\mu}} + \delta(\phi_{<N}) \right)}{\sum_{\phi_N} \exp \left(-\beta J \phi_N \sum_{\mu=1}^d \phi_{k-\hat{\mu}} + \delta(\phi_{<N}) \right)} = \frac{\exp \left(-\beta J \phi_N \sum_{\mu=1}^d \phi_{N-\hat{\mu}} \right)}{\sum_{\phi_N} \exp \left(-\beta J \phi_N \sum_{\mu=1}^d \phi_{N-\hat{\mu}} \right)}$$

So $p(\phi_N | \phi_{<N})$ doesn't depend on all the $N-1$ values within $\phi_{<N}$ but only the d neighbouring values $\phi_{N-\hat{\mu}}$. Proceeding to the next term:

$$\begin{aligned} p(\phi_{N-1} | \phi_{<N-1}) &= \frac{p(\phi_{<N})}{p(\phi_{<N-1})} = \frac{\sum_{\phi_N} \exp \left(-\beta J \phi_N \sum_{\mu=1}^d \phi_{N-\hat{\mu}} + -\beta J \phi_{N-1} \sum_{\mu=1}^d \phi_{N-1-\hat{\mu}} + \delta(\phi_{<N-1}) \right)}{\sum_{\phi_N, \phi_{N-1}} \exp \left(-\beta J \phi_N \sum_{\mu=1}^d \phi_{N-\hat{\mu}} + -\beta J \phi_{N-1} \sum_{\mu=1}^d \phi_{N-1-\hat{\mu}} + \delta(\phi_{<N-1}) \right)} \\ &= \frac{\sum_{\phi_N} \exp \left(-\beta J \phi_N \sum_{\mu=1}^d \phi_{N-\hat{\mu}} + -\beta J \phi_{N-1} \sum_{\mu=1}^d \phi_{N-1-\hat{\mu}} \right)}{\sum_{\phi_N, \phi_{N-1}} \exp \left(-\beta J \phi_N \sum_{\mu=1}^d \phi_{N-\hat{\mu}} + -\beta J \phi_{N-1} \sum_{\mu=1}^d \phi_{N-1-\hat{\mu}} \right)} \end{aligned}$$

and this depends on the neighbours of both ϕ_N and ϕ_{N-1} . In general, we have for $p(\phi_k | \phi_{<N})$:

$$p(\phi_k | \phi_{<N}) = \frac{\sum_{\phi_N, \dots, \phi_{k+1}} \exp \left(-\beta J \sum_{l=k}^N \left(\phi_l \sum_{\mu} \phi_{l-\hat{\mu}} \right) \right)}{\sum_{\phi_N, \dots, \phi_k} \exp \left(-\beta J \sum_{l=k}^N \left(\phi_l \sum_{\mu} \phi_{l-\hat{\mu}} \right) \right)} \quad (7)$$

we can let the outputs of the k^{th} neural network parameterize a mixture of M Gaussians:

$$\{w_j, \mu_j, \sigma_j\}_{j=1}^M = NN_k(B_k)$$

$$p(\phi_k|B_k) \approx \sum_j w_j \mathcal{N}(\mu_j, \sigma_j)$$

which is flexible, as well as easy to sample from. We can exploit the translational invariance of the system, drop the k subscript and sample using the same neural network NN at every position- an approximation that gets better as L gets large. This ensures the number of neural network weights do not scale with system size and also enables a scalable model where a network trained on smaller L can be reused to sample a larger lattice- which should be crucial for lattice field theories where the cost of simulating systems typically scale with the system size. Since the translational invariance is only approximate, we should expect errors to increase when we do an extrapolation to large L . It will be interesting to see if we can engineer the model architecture or the input data to address such sources of error. The log likelihood at every position can be accumulated and optimized using the REINFORCE estimator of the KL divergence between the ansatz and the unnormalized Boltzmann distribution (see [8] for a treatment of the Ising model).

An intriguing alternative would be to model the 2-variable conditional distribution $p(\phi_{k+1}, \phi_k|B_k)$ using a flow-based network like RealNVP[3]. It uses a much more flexible ansatz compared to mixture of Gaussians and *reparameterizable* sampling of the conditionals allows us to optimize the KL divergence directly, and mitigates issues like variance when using the REINFORCE estimator. This would essentially be a compact and scalable version of [1]².

While approaches using autoregressive networks for sampling lattice field theories already exist [5], exploiting the $d-1$ dimensional dependency set means the neural network NN can be a $d-1$ dimensional convolutional network- which can be designed to model stronger dependence on positions closer to ϕ_k than farther ones. A fascinating outcome of lower dimensional inputs is that in the practically important case of $d=4$, it's sufficient to use 3D convolutional layers that have optimized GPU implementations in the CUDA stack or popular deep learning frameworks like PyTorch or Tensorflow- the same are typically absent for 4D convolutions. In a nutshell (and best of my perception), the compact autoregressive network proposed here would be to "local action principle" what convolutional neural networks is to "translationally invariant data". In more general terms, lower-sized inputs helped tackle the curse of dimensionality in gradient-based models and pursuits for them have yielded successful architectures in the Geometric Deep Learning framework[2] and this model could potentially address local action distribution generation with a specialized focus.

²This is an oversimplified picture and there are differences like periodic vs open boundary conditions. Assumption of translational invariance on a finite lattice can contribute to errors which can require more careful model construction to address.

References

- [1] Michael S Albergo, Gurtej Kanwar, and Phiala E Shanahan. “Flow-based generative models for Markov chain Monte Carlo in lattice field theory”. In: *Physical Review D* 100.3 (2019), p. 034515.
- [2] Michael M Bronstein et al. “Geometric deep learning: going beyond euclidean data”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.
- [3] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. “Density estimation using real nvp”. In: *arXiv preprint arXiv:1605.08803* (2016).
- [4] Mathieu Germain et al. “Made: Masked autoencoder for distribution estimation”. In: *International conference on machine learning*. PMLR. 2015, pp. 881–889.
- [5] Di Luo et al. “Gauge invariant autoregressive neural networks for quantum lattice models”. In: *arXiv preprint arXiv:2101.07243* (2021).
- [6] Dinesh PR. *Analysis of Ising model using neural networks*. July 2021. URL: <http://dr.iiserpune.ac.in:8080/xmlui/handle/123456789/6014>.
- [7] Aaron Van den Oord et al. “Conditional image generation with pixelcnn decoders”. In: *Advances in neural information processing systems* 29 (2016).
- [8] Dian Wu, Riccardo Rossi, and Giuseppe Carleo. “Unbiased Monte Carlo cluster updates with autoregressive neural networks”. In: *Physical Review Research* 3.4 (2021), p. L042024.