Variational Monte Carlo and Machine Learning Part II

Artificial Intelligence for Nuclear Physics Winter School
January 12, 2021

Corey Adams & Alessandro Lovato

Argonne NATIONAL LABORATORY

Quick recap

- We consider a non relativistic Hamiltonian of the kind

$$H = \sum_i \frac{\mathbf{p}_i^2}{2m} + \sum_{i<j} v_{ij} + \sum_{i<j<k} V_{ijk} + \dots$$

- Variational Monte Carlo approximately solves the many-body Schrödinger equation assuming a given form of the trial wave function

$$E_T = \frac{\langle \Psi_T | H | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} = \frac{\int dR \langle \Psi_T | R \rangle \langle R | H | \Psi_T \rangle}{\int dR \langle \Psi_T | R \rangle \langle R | \Psi_T \rangle} = \frac{\int dR |\Psi_T(R)|^2 E_L(R)}{\int dR |\Psi_T(R)|^2}$$

- The energy expectation value can be estimated using the central limit theorem

$$\langle E_T \rangle = \frac{1}{N_s} \sum_{R_n} E_L(R_n) \quad \longleftrightarrow \quad E_L(R) = \frac{\langle R | H | \Psi_T \rangle}{\langle R | \Psi_T \rangle}$$

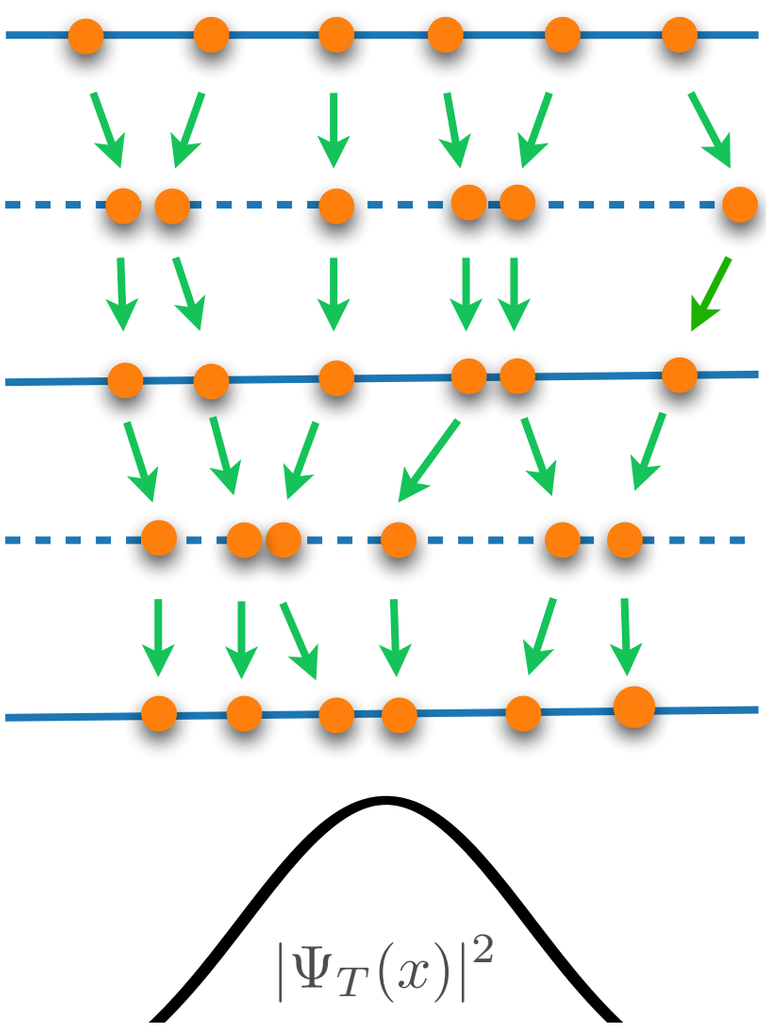
- Where the configurations (walkers) are sampled from

$$P(R) = \frac{|\Psi_T(R)|^2}{\int dR |\Psi_T(R)|^2}$$

Quick recap

- We use the M(RT)^2 algorithm to sample walkers from the distribution

$$P(R) = \frac{|\Psi_T(R)|^2}{\int dR |\Psi_T(R)|^2}$$



- The walkers are sampled from an initial distribution
- Random Gaussian move

$$x_{i+1} = x_i + \zeta$$

- Accept/reject the move according to

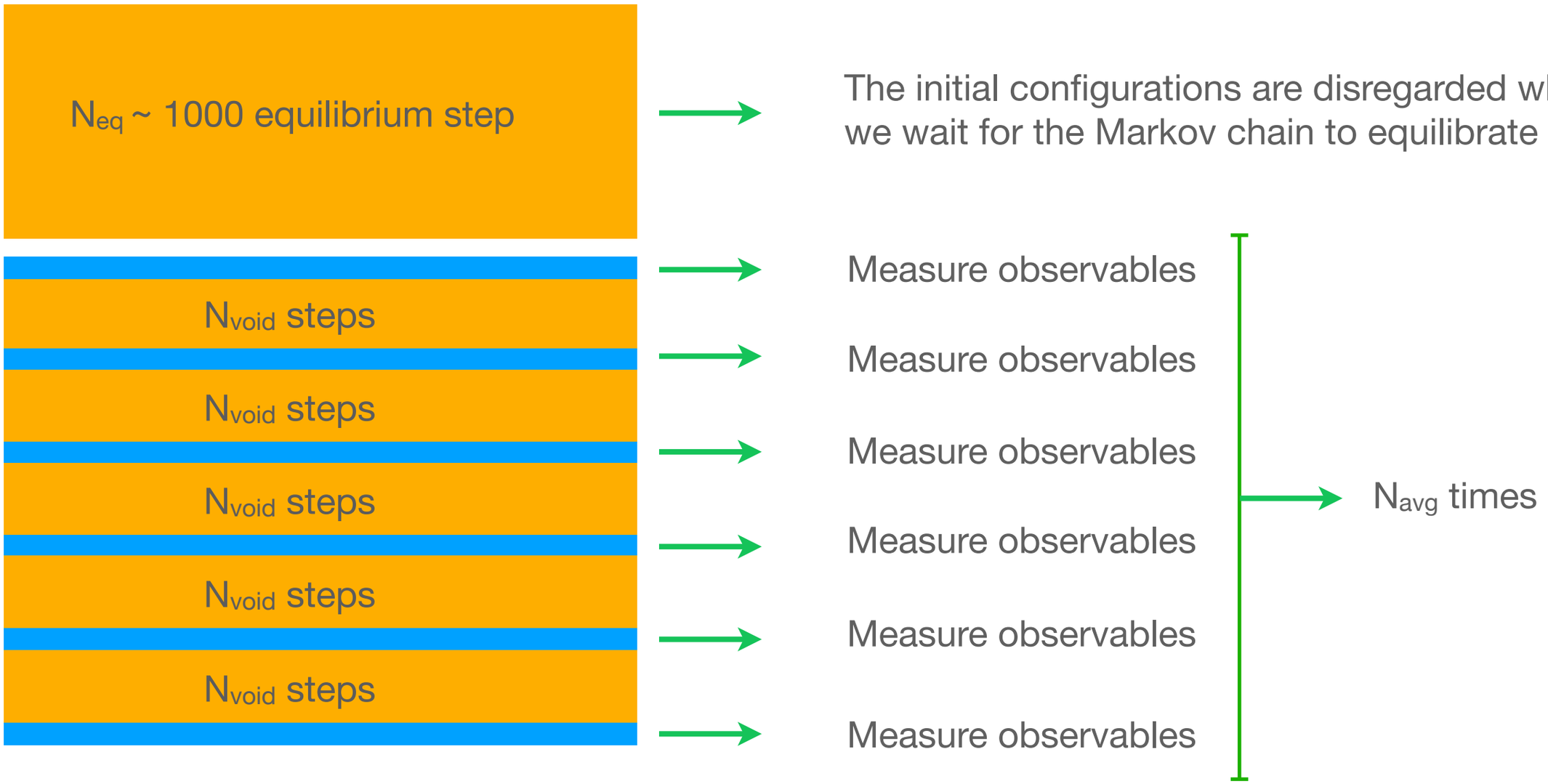
$$\frac{|\Psi_T(y_{i+1})|^2}{|\Psi_T(x_i)|^2} > \xi \quad \rightarrow \quad x_{i+1} = y_{i+1}$$

$$\frac{|\Psi_T(y_{i+1})|^2}{|\Psi_T(x_i)|^2} \leq \xi \quad \rightarrow \quad x_{i+1} = x_i$$

- Iterate until enough configurations are sampled

Quick recap

- Since we use the M(RT)^2 algorithm, some of the configurations that we generate must be disregarded



The quantum Harmonic Oscillator

Let us consider the prototypal problem of a collection of A independent (decoupled) quantum Harmonic oscillators, in N dimensions

$$H = -\frac{1}{2}\sum_{i=1}^{A}\nabla_i^2 + \sum_{i=1}^{A}\frac{\mathbf{r}_i^2}{2}$$

We assume a trial wave function of the form

$$\Psi(R) = \exp\left(-\alpha\sum_{i=1}^{A}\mathbf{r}_i^2\right)$$

So that the exact ground-state wave function is recovered for $\alpha = 1/2$

$$\Psi_0(R) = \exp\left(-\frac{1}{2}\sum_{i=1}^{A}\mathbf{r}_i^2\right) \quad \longleftrightarrow \quad E_0 = A \times N \times \frac{1}{2}$$

The quantum Harmonic Oscillator

The local energy is the sum of the kinetic and potential contributions

$$E_L(R) = \frac{\langle R|H|\Psi_T\rangle}{\langle R|\Psi_T\rangle} = \frac{\langle R|T|\Psi_T\rangle}{\langle R|\Psi_T\rangle} + \frac{\langle R|V|\Psi_T\rangle}{\langle R|\Psi_T\rangle}$$

The kinetic energy involves the second derivative of the trial wave function

$$T_L(R) = -\frac{1}{2}\sum_{i=1}^{A}\frac{\nabla_i^2\Psi_T(R)}{\Psi_T(R)} = -\frac{1}{2}\sum_{i=1}^{A}\left(-2\alpha N + 4\alpha^2\mathbf{r}_i^2\right) = \sum_{i=1}^{A}\left(\alpha N - 2\alpha^2\mathbf{r}_i^2\right)$$

The potential energy is more immediate to evaluate

$$V_L(R) = \frac{1}{2}\sum_{i=1}^{A}\mathbf{r}_i^2$$

Question: What happens for $\alpha = 1/2$?

$$E_L(R) = \frac{1}{2} \times A \times N$$

HO notebook

https://github.com/coreyjadams/AI4NP_School/blob/main/HO_analytic_derivatives.ipynb

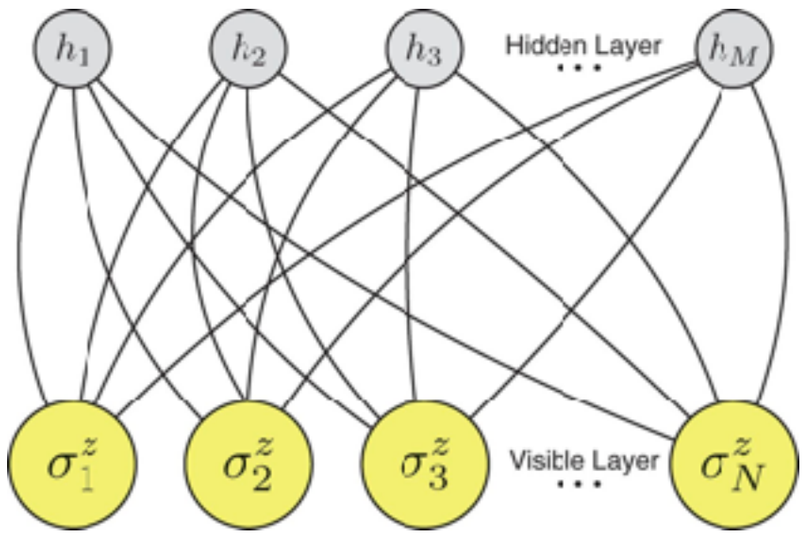
Neural-network quantum states

- Artificial neural networks (ANNs) can compactly represent complex high-dimensional functions;

- Variational representations of **spin-systems** quantum states based on ANNs have been found to outperform conventional variational ansatz;

G. Carleo et al. Science **355**, 602 (2017)

G. Carleo et al. Nat. Commun. **9**, 532 (2018)

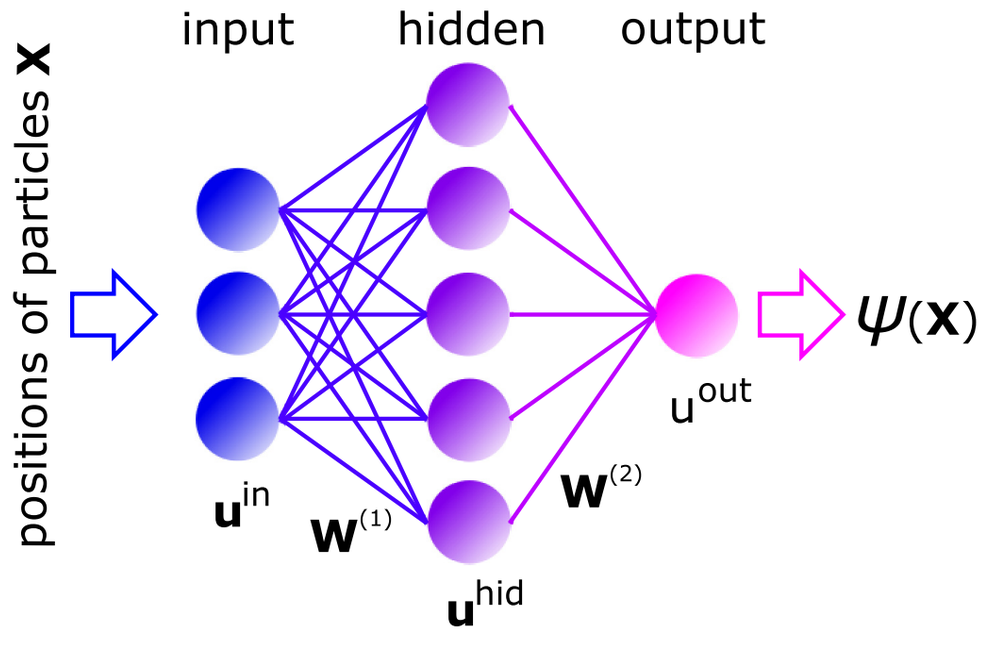


- Applications to the continuum to **few-body systems** and **quantum chemistry** problems have followed shortly thereafter;

H. Saito, J. Phys. Soc. Jpn. **87**, 074002 (2018)

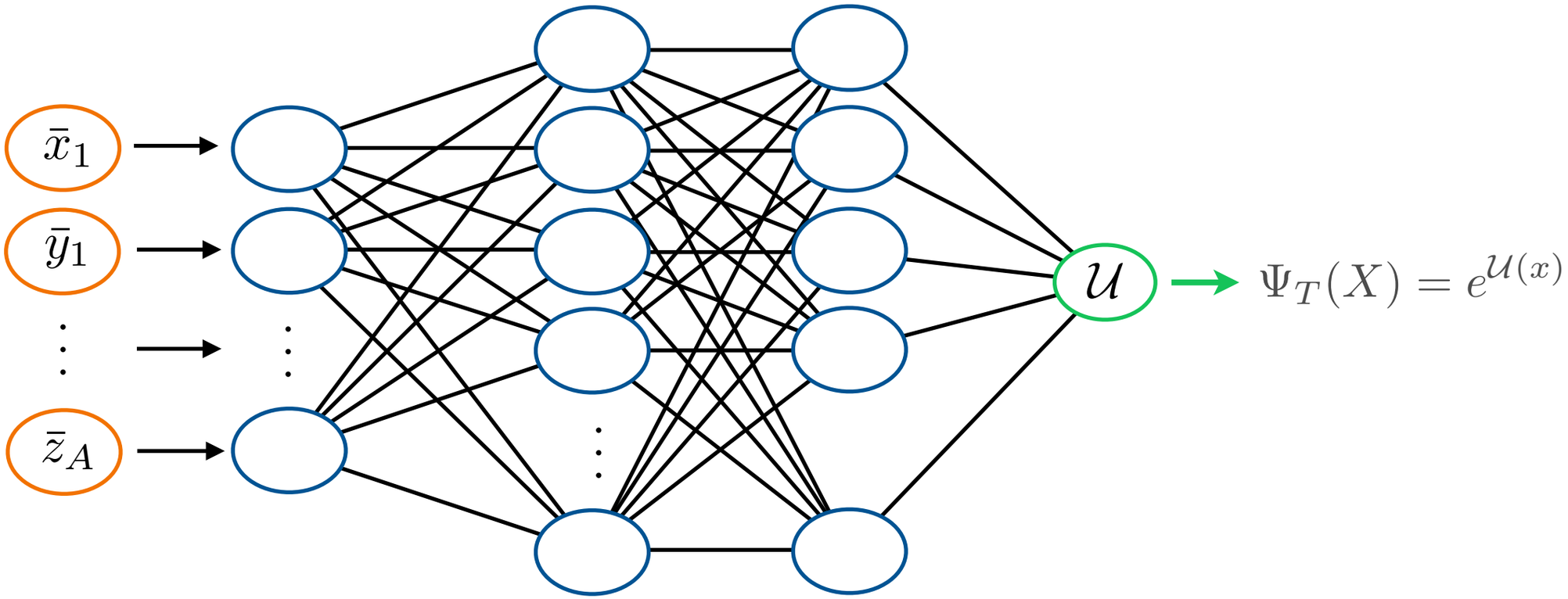
Pfau et al., arXiv:1909.02487 (2019)

Hermann et al., arXiv:1909.08423 (2019)



Neural-network quantum states

- In our examples, we will solve the quantum harmonic oscillator and the hydrogen atom using an ANN representation of the wave function



- The center of mass contributions to the kinetic energy are removed by $\bar{\mathbf{r}}_i = \mathbf{r}_i - \mathbf{R}_{\mathrm{CM}}$

- The kinetic energy requires computing the derivatives of $\mathcal{U}$. We use differentiable **activation functions,** typically sofplus or tanh.

Energy minimization

Minimizing the energy corresponds to training the neural network. Let us recall the derivative of the energy

$$\begin{aligned}\frac{\partial E_T}{\partial p_i} &= 2\left[\frac{\langle\Psi_T|HO^i|\Psi_T\rangle}{\langle\Psi_T|\Psi_T\rangle} - \frac{\langle\Psi_T|H|\Psi_T\rangle}{\langle\Psi_T|\Psi_T\rangle}\frac{\langle\Psi_T|O^i|\Psi_T\rangle}{\langle\Psi_T|\Psi_T\rangle}\right] \\ &= 2\langle HO^i\rangle - \langle H\rangle\langle O^i\rangle \\ &= 2f^i\end{aligned}$$

A Metropolis walk correspond to a “batch” of walkers, and the SGD update reads

$$p_i^{n+1} = p_i^n - \tau\frac{\partial E_T}{\partial p_i} \quad \longrightarrow \quad \tau \simeq 0.001$$

The SGD and its variant (ADAM, RMSprop, momentum…) are greatly successful in training neural networks, but exhibit slow convergence in quantum Monte Carlo applications;

Ultimately, the reason is that sometimes a **small change of the variational parameters correspond to a large change of the wave function**;

Stochastic reconfiguration

We perform an imaginary-time diffusion in the space spanned by the trial wave function and its derivatives

$$(1 - H\tau)|\Psi_T\rangle = \Delta p_0 |\Psi_T\rangle + \sum_i \Delta p_i O^i |\Psi_T\rangle$$

Multiplying from the left by $\langle\Psi_T|/\langle\Psi_T|\Psi_T\rangle$ and $\langle\Psi_T|O^i/\langle\Psi_T|\Psi_T\rangle$ we obtain

$$\begin{cases} \langle(1 - H\tau)\rangle = \Delta p_0 + \sum_i \Delta p_i \langle O^i\rangle \\ \langle O^i(1 - H\tau)\rangle = \Delta p_0 \langle O^i\rangle + \sum_j \Delta p_j \langle O^i O^j\rangle \end{cases}$$

Solving the first line for Δp^0 and inserting back in the second line, we arrive at

$$\Big(\langle H\rangle\langle O^i\rangle - \langle HO^i\rangle\Big)\tau = \sum_j \Delta p_j \Big(\langle O^i O^j\rangle - \langle O^i\rangle\langle O^j\rangle\Big)$$

$$-\frac{1}{2} f_i \tau = \sum_j S_{ij} \Delta p_j$$

S. Sorella, Phys. Rev. B **71**, 241103 (2005)

Stochastic reconfiguration

The stochastic reconfiguration update rule is then given by

$$p_i^{n+1} = p_i^n - \tau \sum_j S_{ij}^{-1} \frac{\partial E_T}{\partial p_j}$$

The SGD is recovered for diagonal S_{ij}, but in general this matrix is not diagonal. This method is a close relative to the “natural gradient approach”.

Simple Gradient: Euclidean distance in the space of parameters

$$ds^2 = \sum_{ij} \delta_{ij} \Delta p_i \Delta p_j$$

Natural Gradient: Riemannian distance in the space of distributions

$$ds^2 = \sum_{ij} S_{ij} \Delta p_i \Delta p_j$$

S. I. Amari, Neural Computation **10**, 251 (1998).

Effectively, the stochastic reconfiguration method “flattens” the space locally and can be considered a 2nd order approach.

Caveat: storing the matrix S_{ij} can be memory consuming for large networks, but the conjugate-gradient method largely overcomes this limitation