Neural-Network Quantum States

A Lecture for the *Machine Learning and Many-Body Physics* workshop

Giuseppe Carleo[1]

June 29 2017, Beijing

[1]Institute for Theoretical Physics, ETH Zurich, Wolfgang-Pauli-Str. 27, 8093 Zurich, Switzerland

Chapter 1

Quantum Mechanics as a Machine Learning Problem

Every machine learning approach has two fundamental ingredients.

1. *The machine:* typically an artificial neural-network, it is a highly dimensional (non-linear) function $F(\mathbf{x}; p_1 \dots p_{N_p})$ of the parameters $p_1 \dots p_{N_p}$
2. *The learning:* the parameters $\mathbf{p}$ are learned on the basis of a stochastic optimization, that minimizes some average loss function $\langle \mathcal{L} \rangle(\mathbf{p})$ on a dataset $\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_{N_s}$. For example $\mathcal{L}(\mathbf{x}_i) = |F(\mathbf{x}_i; \mathbf{p}) - \mathbf{y}_i|$,in the supervised learning setting with expected labels $\mathbf{y}_i$.

On the other hand, the central goal in quantum mechanics is to find a solution to Schroedinger equation

$$H|\Psi_i\rangle \quad = \quad E_i|\Psi_i\rangle, \tag{1.1}$$

for $i = 0, 1, \dots$ and $E_0 < E_1 < \dots$. How can we reduce quantum mechanics then to a machine learning problem?

First of all, I will address the requirement 2, which has been done by pioneers in computational quantum physics like Bill McMillan, in the 60s. Then, I will address requirement 1, which has been done instead only very recently, thus completing the connection between machine learning and quantum mechanics.

1.1 Variational Monte Carlo

To satisfy requirement 2, we need to transform this eigenvalue problem (1.1) into a stochastic optimization problem. To achieve this, we start from an alternative formulation of Schroedinger's equation, based on the variational principle. In particular, consider the energy functional:

$$E[\Psi] = \langle \Psi | H | \Psi \rangle \geq E_0, \tag{1.2}$$

where Ψ is some arbitrary physical state, and E_0 is the exact ground-state energy of the Hamiltonian H. From the variational theorem it is then clear that one can find the

exact ground-state wave-function as the solution of the optimization problem:

$$\Psi_0 \;=\; \mathrm{argmin}_\Psi E[\Psi]. \tag{1.3}$$

For an arbitrary state Ψ, however it is seldom possible to compute analytically the energy functional, since it involves integrals over a high-dimensional space. To solve this problem, in the 60s McMillan realized that the energy functional can be computed *stochastically*. [McMillan1965] In particular, the Variational Monte Carlo method is rooted into the observation that expectation values like (1.2) can be written as statistical averages over a suitable probability distribution.

Let us assume that our Hilbert space is spanned by the many-body kets $|\mathbf{x}\rangle$. These in practice depend on the system in exam. For example in the case of spins 1/2 we would typically have $|\mathbf{x}\rangle = |\sigma_1^z, \sigma_2^z, \dots \sigma_N^z\rangle$, for second-quantized fermions $|\mathbf{x}\rangle = |n_1, n_2, \dots n_N\rangle$, for particles in continuous space $|\mathbf{x}\rangle = |\vec{r}_1, \vec{r}_2, \dots \vec{r}_N\rangle$. The only difference is of course that in the first two cases one has a discrete set of quantum numbers, whereas in the latter case the degrees of freedom are continuos. In both cases we will denote sums over the Hilbert space with discrete sums, although one should always bear in mind that in the case of continuous variables these sums must be interpreted as integrals. In particular we will use the closure relation $\sum_{\mathbf{x}} |\mathbf{x}\rangle\langle\mathbf{x}| = 1$.

1.1.1 Stochastic Estimates of Properties

Using the closure relation, we can rewrite a generic quantum expectation value of some operator O as

$$\frac{\langle\Psi|O|\Psi\rangle}{\langle\Psi|\Psi\rangle} \;=\; \frac{\sum_{\mathbf{x},\mathbf{x}'}\langle\Psi|\mathbf{x}\rangle\langle\mathbf{x}|O|\mathbf{x}'\rangle\langle\mathbf{x}'|\Psi\rangle}{\sum_{\mathbf{x}}\langle\Psi|\mathbf{x}\rangle\langle\mathbf{x}|\Psi\rangle} \tag{1.4}$$

$$= \frac{\sum_{\mathbf{x},\mathbf{x}'}\Psi^\star(\mathbf{x})O_{\mathbf{x}\mathbf{x}'}\Psi(\mathbf{x}')}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2}. \tag{1.5}$$

There can be, in general, two cases:

1. The operator O is diagonal in the computational basis, i.e. $O_{\mathbf{x}\mathbf{x}'} = \delta_{\mathbf{x}\mathbf{x}'}O(\mathbf{x})$. Then

$$\frac{\langle\Psi|O|\Psi\rangle}{\langle\Psi|\Psi\rangle} \;=\; \frac{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2 O(\mathbf{x})}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \tag{1.6}$$

$$\equiv \langle\langle O\rangle\rangle, \tag{1.7}$$

   where $\langle\langle\dots\rangle\rangle$ denote *statistical* expectation values over the probability distribution $\Pi(\mathbf{x}) = |\Psi(\mathbf{x})|^2$. In other words, in this case quantum expectation values are completely equivalent to averaging over Hilbert-space states sampled according to the square-modulus of the wave-function.

2. The operator O is off-diagonal in the computational basis. Then, we can define an auxiliary diagonal operator (often called, in a somehow misleading fashion, *local* operator or estimator)

$$O_{\mathrm{loc}}(\mathbf{x}) \;=\; \sum_{\mathbf{x}'} O_{\mathbf{x}\mathbf{x}'}\frac{\Psi(\mathbf{x}')}{\Psi(\mathbf{x})}, \tag{1.8}$$

such that it is easily proven that

$$\frac{\langle\Psi|O|\Psi\rangle}{\langle\Psi|\Psi\rangle} = \frac{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2 O_{\rm loc}(\mathbf{x})}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \quad (1.9)$$

$$\equiv \langle\langle O_{\rm loc}\rangle\rangle. \quad (1.10)$$

For any observable, then, we can always compute expectation values over arbitrary wave-functions as statistical averages. In the case of off-diagonal operators, it should be noticed that the sum $\sum_{\mathbf{x}'} O_{\mathbf{x}\mathbf{x}'}\frac{\Psi(\mathbf{x}')}{\Psi(\mathbf{x})}$, is extended to the tiny portion of the Hilbert space for which $\mathbf{x}'$ is such that $O_{\mathbf{x}\mathbf{x}'} \neq 0$. For the great majority of physical observables, and for a given $\mathbf{x}$, the number of elements $\mathbf{x}'$ connected by those matrix elements is polynomial in the system size, thus the summation can be carried systematically. This has to be contrasted instead to the summations in $\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2$, where one typically has an exponentially large number of possible values of $\mathbf{x}$ on which to perform the summation, and therefore cannot be done by brute-force. The powerful idea of the Variational Monte Carlo, is therefore to replace these sums over exponentially many states, with a statistical average over a large but finite set of states sampled according to the probability distribution $\Pi(\mathbf{x})$. We therefore have a way to compute, stochastically, the expectation value of *all* the properties of interest. For example we might want to compute the expectation value of σ_i^x for a spin system, the expectation value of $c_i^\dagger c_j$ for fermions, or even the expectation value of the interaction energy $W_{ee}(\vec{r}_1 \ldots \vec{r}_N)$ for our electronic structure problems.

1.1.1.1 Energy

An immediate corollary of the previously presented scheme, is that also the expectation value of the Hamiltonian H (which is itself a generic off-diagonal operator) can be computed using the estimator (1.10). Historically, the local estimator associated to the Hamiltonian is called "local energy":

$$E_{\rm loc}(\mathbf{x}) = \sum_{\mathbf{x}'} H_{\mathbf{x}\mathbf{x}'}\frac{\Psi(\mathbf{x}')}{\Psi(\mathbf{x})}. \quad (1.11)$$

1.2 Stochastic Variational Optimization

The final goal we want to achieve here is to optimize the variational energy. In practice, we assume that the wave function depends on some (possibly millions of) parameters $\mathbf{p} = p_1, \ldots p_M$. We have seen that the expectation value of the energy can be written as a statistical average of the form

$$\langle H\rangle \simeq \langle\langle E_{\rm loc}\rangle\rangle. \quad (1.12)$$

It is easy to show that also the gradient of the energy can be written under to form of the expectation value of some stochastic variable. In particular, define

$$D_k(\mathbf{x}) = \frac{\partial_{p_k}\Psi(\mathbf{x})}{\Psi(\mathbf{x})}, \quad (1.13)$$

then

$$\begin{aligned}
\partial_{p_k}\langle H\rangle &= \partial_{p_k}\frac{\sum_{\mathbf{x},\mathbf{x}'}\Psi^\star(\mathbf{x})H_{\mathbf{x}\mathbf{x}'}\Psi(\mathbf{x}')}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \\
&= \frac{\sum_{\mathbf{x},\mathbf{x}'}\Psi^\star(\mathbf{x})H_{\mathbf{x}\mathbf{x}'}D_k(\mathbf{x}')\Psi(\mathbf{x}')}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2}+\frac{\sum_{\mathbf{x},\mathbf{x}'}\Psi^\star(\mathbf{x})D_k^\star(\mathbf{x})H_{\mathbf{x}\mathbf{x}'}\Psi(\mathbf{x}')}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \\
&\quad -\frac{\sum_{\mathbf{x},\mathbf{x}'}\Psi^\star(\mathbf{x})H_{\mathbf{x}\mathbf{x}'}\Psi(\mathbf{x}')}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2}\frac{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2\left(D_k(\mathbf{x})+D_k^\star(\mathbf{x})\right)}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \\
&= \frac{\sum_{\mathbf{x},\mathbf{x}'}\frac{\Psi^\star(\mathbf{x})}{\Psi^\star(\mathbf{x}')}H_{\mathbf{x}\mathbf{x}'}D_k(\mathbf{x}')|\Psi(\mathbf{x}')|^2+\sum_{\mathbf{x},\mathbf{x}'}|\Psi(\mathbf{x})|^2H_{\mathbf{x}\mathbf{x}'}D_k^\star(\mathbf{x}')\frac{\Psi(\mathbf{x}')}{\Psi(\mathbf{x})}}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \\
&\quad -\langle H\rangle\frac{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2\left(D_k(\mathbf{x})+D_k^\star(\mathbf{x})\right)}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \\
&\simeq \langle\langle E_{\rm loc}D_k^\star\rangle\rangle-\langle\langle E_{\rm loc}\rangle\rangle\langle\langle D_k^\star\rangle\rangle+cc. \qquad (1.14)
\end{aligned}$$

We can therefore compactly write $\partial_{p_k}\langle H\rangle \simeq \langle\langle G_k\rangle\rangle$, with the gradient estimator being

$$G_k(\mathbf{x}) = 2\mathrm{Re}\left[\left(E_{\rm loc}(\mathbf{x})-\langle\langle E_{\rm loc}\rangle\rangle\right)D_k^\star(\mathbf{x})\right]. \qquad (1.15)$$

1.2.1 Zero-Variance Property

One of the most interesting feature of the energy and energy-gradient estimators so-far presented is that they have the so-called zero-variance property: their statistical fluctuations are exactly zero when sampling from the exact ground-state wave-function. Let us consider for example

$$\begin{aligned}
\mathrm{var}(E_{\rm loc}) &= \langle E_{\rm loc}^2\rangle-\langle E_{\rm loc}\rangle^2 \\
&= \sum_{\mathbf{x}}\Psi(\mathbf{x})^2E_{\rm loc}(\mathbf{x})^2-\langle H\rangle^2 \\
&= \sum_{\mathbf{x}}\sum_{\mathbf{x}_1}H_{\mathbf{x},\mathbf{x}_1}\Psi(\mathbf{x}_1)\sum_{\mathbf{x_2}}H_{\mathbf{x},\mathbf{x}_2}\Psi(\mathbf{x}_2)-\langle H\rangle^2 \\
&= \sum_{\mathbf{x}_1}\Psi(\mathbf{x}_1)\sum_{\mathbf{x}}H_{\mathbf{x},\mathbf{x}_1}\sum_{\mathbf{x_2}}H_{\mathbf{x},\mathbf{x}_2}\Psi(\mathbf{x}_2)-\langle H\rangle^2 \\
&= \langle H^2\rangle-\langle H\rangle^2, \qquad (1.16)
\end{aligned}$$

where we have assumed for simplicity that the wave-function is real. Therefore the variance of the local energy is an important physical quantity: the energy variance. It is easy to see that if Ψ is an eigenstate of H then $\langle H^2\rangle=\langle H\rangle^2=E_0^2$, and $\mathrm{var}(E_{\rm loc})=0$, i.e. the statistical fluctuations completely vanish. This property is very important since it also implies that, in a sense to be specified below, the closer we get to the ground-state, the less fluctuations we have on the quantity we want to minimize, the energy.

1.2.2 Stochastic Gradient Descent

The gradient descent method is the simplest optimization scheme, where at each iteration i the variational parameters are modified according to

$$p_k^{i+1} = p_k^i - \eta \partial_{p_k} \langle H \rangle, \tag{1.17}$$

where η is a (small) parameter called the "learning rate" in the machine learning community. An important difference with respect to the non-stochastic (deterministic) gradient descent approach, is that now we only have stochastic averages of the gradient which is therefore subjected to noise. Let us assume for simplicity that all the components of the gradient are subjected to the same amount of gaussian noise with variance σ,i.e.

$$\partial_{p_k} \langle H \rangle = \text{Normal}\left(\langle G_k \rangle, \sigma\right). \tag{1.18}$$

We can then compare Eq. 7 to the discretized Langevin equation:

$$p_k^{i+1} = p_k^i - \delta_t \langle G_k \rangle + \text{Normal}\left(0, \sqrt{2\delta_t T}\right), \tag{1.19}$$

where δ_t is a small time step. This equation samples the Boltzmann distribution

$$\Pi_B(p_1 \dots p_M) = e^{-\frac{\langle H \rangle}{T}}, \tag{1.20}$$

which in the limit $T \to 0$ would converge to the variational ground-state, i.e. to $\min_{\mathbf{p}} \langle H \rangle(p)$. We therefore see that the variance of the gradient corresponds to the effective temperature as

$$\sigma^2 = \text{var}(\bar{G}_k) = 2T/\delta_t \tag{1.21}$$

$$\eta = \delta_t. \tag{1.22}$$

Since we want to find the variational ground state, we should have a scheme in which the temperature is gradually decreased at each optimization step, i.e. $T_1 > T_2 > T_3 \dots$, as in the simulated annealing optimization protocol. The first thing we notice is that $\sigma^2 \simeq \frac{1}{N_s}$, decreases like the number of samples in the Markov chain, therefore

$$T = \frac{\eta \text{var}(\bar{G}_k)}{2} \tag{1.23}$$

$$\propto \frac{\eta}{N_\text{s}} \text{var}(G_k) \tag{1.24}$$

and convenient ways to reduce the temperature are either to reduce the learning rate: $\eta(i) = \eta_0/\sqrt{i+1}$ or to increase the number of samples with the iteration count.

During the optimization however if often happens that if we are close enough to the ground-state solution $\text{var}(G_k) \to 0$. Indeed, it is easy to show that for an exact eigenstate the statistical fluctuations of the gradient are exactly vanishing, i.e. $\text{var}(G_k) = 0$. In practice then, even a constant number of samples and a fixed (small) η are sufficient to converge to the ground-state, provided that one checks during the optimization that the value of the effective temperature (1.23) is actually going to zero as expected.

1.3 Example: Jastrow Factors

Let us give a specific example of variational states, we consider now a system of interacting particles in continuous space, for which the most general Hamiltonian is

$$H = -\frac{\hbar^2}{2m}\sum_i^N \nabla^2_{\vec{r}_i} + \sum_i V_1(\vec{r}_i) + \sum_{i<j} V_2(\vec{r}_i,\vec{r}_j), \tag{1.25}$$

where V_1 and V_2 are generic one and two-body interaction potential.

We now define the exact Jastrow-Feenberg expansion for the many-body state:

$$\Psi_p(\vec{r}_1,\dots\vec{r}_N) = \Psi_0(\vec{r}_1,\dots\vec{r}_N)\times\exp\left[\sum_i J_1(\vec{r}_i) + \frac{1}{2}\sum_{i\neq j} J_2(\vec{r}_i,\vec{r}_j) + \right.$$
$$\left. + \dots \frac{1}{p!}\sum_{i_1\neq i_2\neq\dots i_p} J_p(\vec{r}_{i_1},\vec{r}_{i_2}\dots\vec{r}_{i_p})\right], \tag{1.26}$$

where $\Psi_0(\vec{r}_1,\dots\vec{r}_N)$ is some parameter-independent wave-function, and the variational parameters are the functions $J_1(\vec{r})$,$J_2(\vec{r},\vec{r}')$,... $J_p(\vec{r}_1,\vec{r}_2\dots\vec{r}_p)$. These expansion is clearly exact when $p=N$, however in practice one observes convergence to the exact ground-state much sooner, and typically $p=2,3$ are enough to obtain very accurate results.

1.3.1 One-dimensional trapped particles

As a simple exercise one can consider single-particle, one-dimensional Hamiltonians of trapped particles, for which $V_1(x)$ is an even function of x and $V_2=0$ (non-interacting particles). In this case (for symmetry reasons) one can write the function expansion $J_1(x) = p_1x^2 + p_2x^4 + \dots$, where p_1 ,p_2 etc are the parameters to be determined variationally

$$\frac{1}{\Psi(x)}\frac{\partial^2}{\partial x^2}\Psi(x) = J_1''(x) + J_1'(x)^2. \tag{1.27}$$

In this case it is easy to show that

$$E_{\rm loc}(x) = -\frac{\hbar^2}{2m}\left(J_1''(x) + J_1'(x)^2\right) + V_1(x) \tag{1.28}$$
$$D_k(x) = x^{2k}. \tag{1.29}$$

Chapter 2

Neural-Network Quantum States

In the first part of this lecture we have rephrased the problem of finding a ground state in terms of a stochastic optimization problem. To really take advantage of the potentialities of machine learning, however it is still necessary to accomplish task 1, in our previous list: we need to define a suitable machine to solve our learning problem. This is what we have done in our recent work. [CarleoTroyer2017]

2.1 Wave-Function as a Neural Network

The fundamental problem with the stochastic optimization problem described before is that, in principle, to achieve the exact ground state energy one needs to consider exponentially many parameters. To see this point, consider the case of N spin 1/2 particles, then the exact ground-state wave-function is fully specified by the 2^N amplitudes

$$\langle \mathbf{x}|\Psi\rangle \quad = \quad \Psi(\mathbf{x}), \tag{2.1}$$

for all the possible values of $\mathbf{x} = \sigma_1^z \sigma_2^z \dots \sigma_N^z$.

This task however is clearly unfeasible when the number of particles N is too large. For example, one can do a back-of-the-envelope calculation to show that only *storing* the wave-function for more than 100 spins would require a number of atoms larger than what can be found on our planet!

However, this exponential complexity is not necessary a limiting factor. In this case we can indeed think of using the ability of artificial neural networks to compress high-dimensional data into a low-dimensional representation.

The starting point is to ask a suitable neural network to *compute* the wave-function amplitudes. Formally, we then set:

$$\Psi(\mathbf{x}) = F(\mathbf{x}; p_1, p_2 \dots p_{N_p}), \tag{2.2}$$

where F is the output of a suitably chosen artificial neural network, depending on a set of parameters $\mathbf{p}$.

2.2 Restricted Boltzmann Machines

The choice of the specific neural network used to represent the wave-function is arbitrary, provided that it is reasonably expressive (i.e. that in the limit of large N_p we can always recover the exact wave-function).

A convenient choice is the so-called Restricted Boltzmann Machine (RBM), which is defined as:

$$F_{\rm rbm}(\sigma_1^z, \sigma_2^z, \dots \sigma_N^z) = \sum_{\{h\}} \exp\left[\sum_{ij} W_{ij}\sigma_i^z h_j + \sum_j h_j b_j + \sum_i \sigma_i^z a_i\right], \quad (2.3)$$

where the network parameters are W,a, and b. This architecture corresponds to the partition function of a gas of M hidden units (h_j) connected to the physical spins (σ_i^z). Since the connections are allowed only between hidden and visible units, but not between hidden units, nor between visible units, this architecture is called *restricted.* Because of this restriction, however it is easy to compute F explicitly. Indeed

$$\sum_{\{h\}} \exp\left[\sum_{ij} W_{ij}\sigma_i^z h_j + \sum_j h_j b_j + \sum_i \sigma_i^z a_i\right] = \quad (2.4)$$

$$e^{\sum_i \sigma_i^z a_i} \times \sum_{\{h\}} \Pi_j \exp\left[\sum_i W_{ij}\sigma_i^z h_j + h_j b_j\right] = \quad (2.5)$$

$$e^{\sum_i \sigma_i^z a_i} \times \Pi_j \left(\exp\left[\sum_i W_{ij}\sigma_i^z + b_j\right] + \exp\left[-\sum_i W_{ij}\sigma_i^z - b_j\right]\right) = \quad (2.6)$$

$$e^{\sum_i \sigma_i^z a_i} \times \Pi_j 2\cosh\left[\sum_i W_{ij}\sigma_i^z + b_j\right] . \quad (2.7)$$

Because the wave-function, in general, can be complex valued, also the weights in this expression should be taken complex. It is easy to convince one-self that if this is the case than the wave-function takes arbitrary complex values.

2.3 An example implementation

During the lecture I will show an example implementation of the stochastic optimization algorithm for neural-network RBM states. In particular, I will consider the transverse-filed Ising hamiltonian in 1D:

$$H = -h\sum_i \sigma_i^x - J\sum_i \sigma_i^z \sigma_{i+1}^z, \quad (2.8)$$

with periodic boundary conditions over a ring of L sites. To simplify things, and knowing that the ground-state wave-function in this case is positive definite, I will consider the following quantum state [TGC2017]:

$$\Psi(\sigma_1^z, \sigma_2^z, \dots \sigma_N^z) = \sqrt{F_{\rm rbm}(\sigma_1^z, \sigma_2^z, \dots \sigma_N^z)}, \quad (2.9)$$

where the specific RBM taken here contains only real-valued parameters. An advantage of this formulation is that sampling from $|\Psi(\sigma_1^z, \sigma_2^z, \ldots \sigma_N^z)|^2 = F_{\rm rbm}(\sigma_1^z, \sigma_2^z, \ldots \sigma_N^z)$ is particularly easy, since it can been done using alternate Gibbs sampling.

2.3.1 Gibbs Sampling

Gibbs sampling is a special case of the Metropolis-Hastings algorithm (see Appendix A). When the RBM has only real-valued parameters, then one can interpret the quantity

$$P(\sigma, h) \;=\; \exp\left[\sum_{ij} W_{ij}\sigma_i^z h_j + \sum_j h_j b_j + \sum_i \sigma_i^z a_i\right], \tag{2.10}$$

as a joint probability density (apart from a global normalization) of the physical and hidden units. [FischerIgel2014]

The idea of alternate Gibbs sampling is then to devise a two step Markov-chain sampling with transition probabilities:

$$T_\sigma((\sigma, h) \to (\sigma', h)) = \frac{P(\sigma', h)}{\sum_{\sigma''} P(\sigma'', h)} = P(\sigma'|h) \tag{2.11}$$

$$T_h((\sigma, h) \to (\sigma, h')) = \frac{P(\sigma, h')}{\sum_{h''} P(\sigma, h'')} = P(h'|\sigma). \tag{2.12}$$

The acceptance probability for these two type of moves can be readily computed using the Metropolis-Hasting acceptance rule:

$$A(\mathbf{x} \to \mathbf{x}') \;=\; \min\left(1, \frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x})} \times \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')}\right), \tag{2.13}$$

where in one case spin configuration are changed, $\mathbf{x}' = (\sigma', h)$ and in the other case hidden variable configuration are changed, thus $\mathbf{x}' = (\sigma, h')$. For example, in the first case the acceptance probability reads:

$$\begin{aligned} A((\sigma, h) \to (\sigma', h)) &= \min\left\{1, \frac{P(\sigma', h)}{P(\sigma, h)} \times \frac{T_\sigma((\sigma', h) \to (\sigma, h))}{T_\sigma((\sigma, h) \to (\sigma', h))}\right\} \\ &= \min\left\{1, \frac{P(\sigma', h)}{P(\sigma, h)} \times \frac{P(\sigma|h)}{P(\sigma'|h)}\right\} \\ &= 1, \end{aligned} \tag{2.14}$$

where in the last line we have used the fact that

$$\frac{P(\sigma|h)}{P(\sigma'|h)} \;=\; \frac{\frac{P(\sigma,h)}{\sum_{\sigma''} P(\sigma'',h)}}{\frac{P(\sigma',h)}{\sum_{\sigma''} P(\sigma'',h)}} = \frac{P(\sigma, h)}{P(\sigma', h)}. \tag{2.15}$$

The same reasoning can be done for moves that change the hidden units only, and one gets an acceptance of 1 as well. The important point is that $P(\sigma|h)$ and $P(h|\sigma)$ can be computed exactly for an RBM.

For example, we have that:

$$\begin{aligned} P(h|\sigma) &= \frac{P(\sigma,h)}{\sum_{h''} P(\sigma,h'')} \\ &= \frac{e^{\sum_i \sigma_i^z a_i} \times \Pi_j \exp\left[\sum_i W_{ij}\sigma_i^z h_j + b_j h_j\right]}{e^{\sum_i \sigma_i^z a_i} \times \Pi_j 2\cosh\left[\sum_i W_{ij}\sigma_i^z + b_j\right]}, \end{aligned} \tag{2.16}$$

and each hidden variable has a probability which is independent on the value of the other hidden variables. We have:

$$P(h_j = 1|\sigma) = \text{Logistic}(2\theta_\text{j}) \tag{2.17}$$

$$P(h_j = -1|\sigma) = \text{Logistic}(-2\theta_\text{j}), \tag{2.18}$$

where $\text{Logistic}(x) = \frac{1}{1+\exp(-x)}$ and $\theta_j = \sum_i W_{ij}\sigma_i^z + b_j$. A similar expression can be derived also for the other conditional probability, which reads:

$$\begin{aligned} P(h|\sigma) &= \frac{P(\sigma,h)}{P(h)} \\ &= \frac{e^{\sum_j h_j b_j} \times \Pi_i \exp\left[\sum_j W_{ij}\sigma_i^z h_j + a_i\sigma_i\right]}{e^{\sum_j h_j b_j} \times \Pi_i 2\cosh\left[\sum_j W_{ij} h_j^z + a_i\right]}, \end{aligned} \tag{2.19}$$

and each spin variable has a probability which is independent on the value of the other spin variables. We then have:

$$P(\sigma_i = 1|h) = \text{Logistic}(2\gamma_\text{i}) \tag{2.20}$$

$$P(\sigma_i = -1|h) = \text{Logistic}(-2\gamma_\text{i}), \tag{2.21}$$

where $\gamma_i = \sum_j W_{ij} h_j^z + a_i$. Proposing spin and hidden variable configurations according to the Gibbs transition probability is therefore very easy and consists in the following:

1. Generate N random numbers $r_i \in [0, 1)$.

2. Set the i-th spin with probability $P(\sigma_i = 1|h) = \text{Logistic}(2\gamma_\text{i})$, i.e. if $P(\sigma_i = 1|h) > r_i$ then set $\sigma_i' = 1$ otherwise $\sigma_i' = -1$.

3. Generate M random number $l_j \in [0, 1)$.

4. Set the j-th hidden unit with probability $P(h_j = 1|\sigma) = \text{Logistic}(2\theta_\text{j})$, i.e. if $P(h_j = 1|\sigma) > l_j$ then set $h_j' = 1$ otherwise $h_j' = -1$.

Repeating these steps N_s times, we then generate spin configurations which are sampled from $|\Psi(\sigma_1^z, \sigma_2^z, \dots \sigma_N^z)|^2 = F_\text{rbm}(\sigma_1^z, \sigma_2^z, \dots \sigma_N^z)$ –Eq. (2.3). Notice that this scheme is rather easy to implement since we do not need to perform a Metropolis-Hastings test at each step of the Markov chain, given that all moves are accepted (see Appendix A for details).

2.3.2 Computing the local energy

For a given spin configuration, we also need to compute the local energy:

$$E_{\rm loc}(\sigma) \;=\; \sum_{\sigma'} H_{\sigma,\sigma'} \frac{\psi(\sigma')}{\psi(\sigma)}. \tag{2.22}$$

For the transverse-field Ising model, the sum runs over the $N+1$ configurations $\sigma'(0) = \sigma$ and $\sigma'(k) = \sigma_1^z \cdots - \sigma_k^z \ldots \sigma_N^z$, with $H_{\sigma,\sigma} = -J \sum_i \sigma_i^z \sigma_{i+1}^z$ and $H_{\sigma,\sigma'(k>0)} = -h$. This sum can then be computed in polynomial time, and it is efficiently done pre-computing the values of the "angles" θ_j. In particular,

$$\frac{\psi(\sigma'(k))}{\psi(\sigma)} \;=\; e^{-2a_k\sigma_k} \times \Pi_j \frac{\cosh(\theta_j - 2\sigma_k W_{jk})}{\cosh(\theta_j)}. \tag{2.23}$$

2.3.3 Computing the variational derivatives

The variational derivatives

$$D_k(\sigma) \;=\; \frac{\partial_{p_k}\Psi(\sigma)}{\Psi(\sigma)}, \tag{2.24}$$

can also be computed efficiently and read:

$$D_{a_i}(\sigma) \;=\; \frac{1}{2}\sigma_i^z, \tag{2.25}$$

$$D_{b_j}(\sigma) \;=\; \frac{1}{2}\tanh(\theta_j), \tag{2.26}$$

$$D_{W_{ij}}(\sigma) \;=\; \frac{1}{2}\tanh(\theta_j)\sigma_i^z. \tag{2.27}$$

Bibliography

[McMillan1965] William L. McMillan, Phys. Rev. 138, A442 (1965)

[FischerIgel2014] Asja Fischer, and Christian Igel. "Training Restricted Boltzmann Machines: An Introduction". *Pattern Recognition* 47, 25-39, 2014.

[CarleoTroyer2017] Giuseppe Carleo, and Matthias Troyer. "Solving the quantum many-body problem with artificial neural networks". *Science* 355, 602-606, 2017.

[TGC2017] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. *arXiv* 1703.05334, 2017.

Appendix A

Sampling Methods

During the lecture we have established a fundamental connection between quantum mechanics and statistical sampling. For this mapping to be efficient, we need an efficient way of sampling from the probability distribution $\Pi(\mathbf{x}) = |\Psi(\mathbf{x})|^2$. In particular the goal is to generate N_s samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots \mathbf{x}^{(N_s)}$ such that we can estimate expectation values as averages over those samples:

$$\langle\langle O_{\text{loc}}\rangle\rangle \simeq \frac{1}{N_s}\sum_i O_{\text{loc}}(\mathbf{x}^{(i)}). \tag{A.1}$$

A.0.1 Markov Chain and Detailed Balance

A Markov chain is completely specified by the transition probability $\mathcal{T}(\mathbf{x}^{(i)} \to \mathbf{x}^{(i+1)})$, i.e. given a sample $\mathbf{x}^{(i)}$, we transition to the next element of the chain with probability T. The transition probability (as all well-define probabilities) must always be normalized: $\sum_{\mathbf{x}'} \mathcal{T}(\mathbf{x} \to \mathbf{x}') = 1$.

We would like to devise a Markov chain process such that $\Pi^{\text{mc}}(\mathbf{x}) = \Pi(\mathbf{x})$, i.e. that the probability with which a given state $\mathbf{x}$ appears in the chain is equal to desired probability we want to sample from.

An important condition for this to happen is that the probability distribution $\Pi^{\text{mc}}(\mathbf{x})$ is *stationary*, i.e. all states along the chain should be distributed according to the same probability, and this should not change along the chain. A sufficient condition for this to happen is that

$$\Pi(\mathbf{x})\mathcal{T}(\mathbf{x} \to \mathbf{x}') = \Pi(\mathbf{x}')\mathcal{T}(\mathbf{x}' \to \mathbf{x}), \tag{A.2}$$

which is called *detailed balance* equation. This condition basically enforces stationarity (also called micro-reversibility) in the chain: the probability of being in a given state $\mathbf{x}$ and of doing a transition to another state $\mathbf{x}'$ must be equal to the reverse process, starting from $\mathbf{x}'$ and transitioning to $\mathbf{x}$.

A.0.2 The Metropolis-Hastings Algorithm

There exist many possible transition probabilities that satisfy the detailed balance condition (A.2), however the most famous choice is certainly the Metropolis-Hastings pre-

scription. In this case, we separate the transition process into two steps:

$$\mathcal{T}(\mathbf{x} \to \mathbf{x}') \;=\; T(\mathbf{x} \to \mathbf{x}')A(\mathbf{x} \to \mathbf{x}'), \tag{A.3}$$

i.e. we first propose a state with some (simple) probability distribution $T(\mathbf{x} \to \mathbf{x}')$ we can easily sample from, and then accept or reject the new state $\mathbf{x}'$ as the next element of the chain with probability $A(\mathbf{x} \to \mathbf{x}')$.

Using the detailed balance condition, we see that the acceptance probability must satisfy:

$$\frac{A(\mathbf{x} \to \mathbf{x}')}{A(\mathbf{x}' \to \mathbf{x})} \;=\; \frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x})} \times \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')}. \tag{A.4}$$

A possible acceptance that satisfies this condition is:

$$A(\mathbf{x} \to \mathbf{x}') \;=\; \min\left(1, \frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x})} \times \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')}\right). \tag{A.5}$$

Notice that this acceptance probability satisfies (A.4), since if $\frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x})} \times \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')} < 1$ then $\frac{\Pi(\mathbf{x})}{\Pi(\mathbf{x}')} \times \frac{T(\mathbf{x} \to \mathbf{x}')}{T(\mathbf{x}' \to \mathbf{x})} > 1$, $A(\mathbf{x}' \to \mathbf{x}) = 1$ and (A.4) is trivially verified. The same reasoning can be applied for the case $\frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x})} \times \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')} > 1$.

The Metropolis-Hasting Algorithm can be then summarized in the following steps:

1. Generate a random state $\mathbf{x}'$ drawing from the (simple) transition probability $T(\mathbf{x}^{(i)} \to \mathbf{x}')$.

2. Compute the quantity

$$R \;=\; \frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x}^{(i)})} \times \frac{T(\mathbf{x}' \to \mathbf{x}^{(i)})}{T(\mathbf{x}^{(i)} \to \mathbf{x}')}. \tag{A.6}$$

3. Draw a uniformly distributed random number $\eta \in [0, 1)$.

4. If $R > \eta$, accept the new states, i.e. $\mathbf{x}^{(i+1)} = \mathbf{x}'$. Otherwise, the following state in the chain stays the current one: $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)}$.

Notice that steps 2-4 are necessary to decide whether to accept or reject the proposed state according to the Metropolis probability (A.5).

Appendix B

Estimating Errors and Auto-Correlation Times

Since Markov chains are generated transitioning from a state to the next one, it is natural to expect that adjacent points in the chain will be statistically correlated. To quantify this notion of correlation more precisely, let us first consider the Markov chain estimate for the expectation value of a given function:

$$\bar{g}_{n_s} = \frac{1}{n_s}\sum_{i}^{n_s} g_i, \tag{B.1}$$

where we have used the short-hand $g_i \equiv g(\mathbf{x}^{(i)})$. The law of large numbers states that

$$\bar{g}_{n_s} \overrightarrow{n_s \to \infty} \sum_{\mathbf{x}} \Pi(\mathbf{x}) g(\mathbf{x}), \tag{B.2}$$

and the central limit theorem says that $\bar{g}_{n_s}$ is a random variable normally distributed,

$$\text{Prob}(\bar{g}_{n_s}) = \text{Normal}(\bar{\text{g}}_\infty, \sigma^2), \tag{B.3}$$

with expected value $\bar{g}_\infty$ and variance $\sigma^2 = \text{var}(\bar{g}_{n_s})$, where the variance is computed over different realizations of the Markov chain. It explicitly reads

$$\begin{aligned}
\text{var}(\bar{g}_{n_s}) &= \text{var}\left(\frac{1}{n_s}\sum_i g_i\right) \\
&= \frac{1}{n_s^2}\sum_{ij}\langle g_i g_j\rangle - \frac{1}{n_s^2}\sum_{ij}\langle g_i\rangle\langle g_j\rangle \\
&= \frac{1}{n_s}\left(\frac{1}{n_s}\sum_i\left(\langle g_i^2\rangle - \langle g_i\rangle^2\right) + \frac{2}{n_s}\sum_i\sum_{j=i+1}\left(\langle g_i g_j\rangle - \langle g_i\rangle\langle g_j\rangle\right)\right) \\
&= \frac{1}{n_s}\text{var}(g_0) + 2\sum_{j=1}^{n_s}\left(\langle g_0 g_j\rangle - \langle g_0\rangle\langle g_j\rangle\right)\left(1 - \frac{j}{n_s}\right),
\end{aligned} \tag{B.4}$$

where we assumed that the Markov chain is stationary, i.e. $\text{var}(g_i)$ does not depend on the index i and the same for the covariance. Therefore

$$\text{var}(\bar{g}_{n_s}) = \frac{1}{n_s}\text{var}(g_0) 2\tau_{\text{int}}, \tag{B.5}$$

having defined the integrated auto-correlation time as

$$\tau_{\rm int} = \frac{1}{2} + \sum_{j=1}^{n_s} \left(\langle g_0 g_j \rangle - \langle g_0 \rangle \langle g_j \rangle \right) \left(1 - \frac{j}{n_s} \right). \tag{B.6}$$

We therefore see that unless the Markov chain samples are completely uncorrelated (i.e. $\langle g_s g_j \rangle - \langle g_s \rangle \langle g_j \rangle = 0$) the statistical error on the estimator $\bar{g}_{n_s}$ is increased by the positive factor $\tau_{\rm int}$.

A way to correctly estimate the integrated autocorrelation time is through the correlation function

$$\rho(j) = \frac{\langle g_0 g_j \rangle - \langle g \rangle^2}{\langle g^2 \rangle - \langle g \rangle^2}, \tag{B.7}$$

and a numerically stable estimate of the correlation time is given by

$$\tau_{\rm int} \simeq \frac{1}{2} + \sum_{j=1}^{j_{\rm cut}} \rho(j), \tag{B.8}$$

where $j_{\rm cut}$ is chosen for numerical stability as the first j such that $\rho(j_{\rm max}) < 0$. In practice, given a sequence of estimates $g_1, \ldots g_{n_s} = \mathbf{g}$, then the correlation function can be efficiently estimated with a sequence of Fast Fourier Transforms and its inverses:

$$A = FFT(\mathbf{g} - \bar{g}), \tag{B.9}$$

$$B = AA^\star, \tag{B.10}$$

$$\rho = \frac{FFT^{-1}(B)}{\langle g^2 \rangle - \langle g \rangle^2}. \tag{B.11}$$