On the Thermodynamic Interpretation of Deep Learning Systems

Rita Fioresi[1(✉)], Francesco Faglioni[2], Francesco Morri[3], and Lorenzo Squadrani[1]

[1] University of Bologna, Bologna, Italy
rita.fioresi@unibo.it, lorenzo.squadrani@studio.unibo.it
[2] University of Modena, Modena, Italy
francesco.faglioni@unimore.it
[3] Politecnico di Torino, Torino, Italy
francesco.morri@studenti.polito.it

Abstract. In the study of time evolution of the parameters in Deep Learning systems, subject to optimization via SGD (stochastic gradient descent), temperature, entropy and other thermodynamic notions are commonly employed to exploit the Boltzmann formalism. We show that, in simulations on popular databases (CIFAR10, MNIST), such simplified models appear inadequate: different regions in the parameter space exhibit significantly different temperatures and no elementary function expresses the temperature in terms of learning rate and batch size, as commonly assumed. This suggests a more conceptual approach involving contact dynamics and Lie Group Thermodynamics.

Keywords: Deep Learning · Statistical mechanics · Lie groups machine learning

1 Introduction

In the study of artificial neural networks, thermodynamics and statistical mechanics modeling proved to be a driving force leading to the development of new algorithms, starting from the pioneering work by Jaynes [8], going from Hopfield neural networks [7] to Boltzmann machines [1] and their newer restricted and deep versions [12]. As the new successful family of *Deep Learning* algorithms emerged, the language of thermodynamics and statistical mechanics is commonly employed to draw analogies and boost intuition on the functioning of optimizers based on SGD (Stochastic Gradient Descent), see [5,6] and refs. therein.

Our purpose is to establish a dictionary connecting neural networks notions commonly used in such algorithms (e.g. loss, parameters, learning rate, mini-batch, etc.) and statistical mechanics concepts (e.g. particles, masses, energy, temperature, etc.), so that the analogies may be exploited with a deeper understanding and go beyond a qualitative analysis.

F. Nielsen and F. Barbaresco (Eds.): GSI 2021, LNCS 12829, pp. 909–917, 2021.
https://doi.org/10.1007/978-3-030-80209-7_97

Our paper is organized as follows. In Sect. 2 we establish the above mentioned correspondence, relating thermodynamics concepts with neural networks ones. We then validate our model in Sect. 3 through some experiments and in Sect. 4 we suggest a more conceptual approach based on the formalism of contact dynamics and Lie Groups Thermodynamics [3,4,13].

2 Thermodynamics and Deep Learning

Let $\Sigma = \{z_i \,|1 \leq i \ \leq N\} \subset \mathbb{R}^D$ represent a dataset of size N, i. e. $|\Sigma| = N$ and let $f = \frac{1}{N}\sum_{i=1}^{N} f_i$ be the loss function, f_i being the loss of the i-th datum z_i. A popular choice for f, for example, is the Kullback-Leibler divergence of the Amari loss [2] (Softmax). We assume the training to take place through Stochastic Gradient Descent (SGD) with minibatch $\mathcal{B}$, $|\mathcal{B}| << N$. We call $\overline{x} \in \mathbb{R}^d$ the vector consisting of the learning parameters of the model. The value of the k^{th} parameter is thus x_k. Since parameters evolve in time during training, we write $\overline{x}(t)$, or $x_k(t)$ to emphasize this. In practical implementations, with optimization obtained via Gradient Descent (GD), t is a discrete variable indicating the timestep:

$$\overline{x}(t+1) = \overline{x}(t) - \eta \nabla f \tag{1}$$

η denoting the learning rate. Equation (1) is often expressed in its continuous version as:

$$\frac{d\overline{x}}{dt} = -\eta \nabla f \tag{2}$$

Notice that, since Σ is fixed, the loss function $f(t)$ at time t is determined by the parameters $\overline{x}(t)$. If SGD is used for optimization, Eq. (2) could be substituted with (see [6]):

$$\frac{d\overline{x}}{dt} = -\eta \nabla f_{\mathcal{B}} \tag{3}$$

where the full loss function is replaced with $f_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} f_i$ and at each time step $\mathcal{B}$ is chosen in Σ. Our purpose is to show that if (2) is properly interpreted in a thermodynamics context, we can analyze effectively the dynamics of SGD, without introducing stochastic variables as in (3) studied in [6], besides the ones intrinsic to Boltzmann statistical mechanics and its far reaching generalizations (see [3,4,11] and refs. therein).

We now proceed with our thermodynamic interpretation. Let $\overline{x}(t)$ represent the position (or geometry) of a mechanical system at time t and consider the loss $f(t)$ as the potential energy associated with the geometry of the system.

We first look at the system as conservative, i.e., the force acting on each particle according to (2) is the negative gradient of the potential:

$$F_k = -\frac{\partial f}{\partial x_k} \qquad \text{or} \qquad \overline{F} = -\nabla f$$

The velocity at each optimization step is:

$$v_k(t) = \frac{x_k(t) - x_k(t-1)}{\Delta t} \qquad \text{or} \qquad \overline{v}(t) = \frac{\overline{x}(t) - \overline{x}(t-1)}{\Delta t}$$

We assign masses m_k to each particle. i.e. to each parameter x_k. If the mechanical system were ideal and isolated, it would evolve following Newton's law:

$$F_k = m_k \frac{dv_k}{dt}$$

For finite time steps Δt, the corresponding position update would be

$$\Delta x_k = v_k \Delta t + \frac{1}{2} \frac{F_k}{m_k} (\Delta t)^2 \tag{4}$$

In this case, the total energy would be conserved (up to numerical integration errors) and the system would convert potential into kinetic energy and vice versa.

In the language of atomistic simulations, this is referred to as *Constant Energy* dynamics, but it is *not* what occurs during neural network training. In fact, typical optimization algorithms use the gradient to update coordinates, not velocities. The equation ruling the dynamics is in fact (1): $\Delta \overline{x} = -\eta \nabla f \, \Delta t$. Let us rewrite (4), taking the force term as the gradient of the potential:

$$\Delta x_k = v_k \Delta t - \frac{\Delta t}{2 m_k} \frac{\partial f}{\partial x_k} \Delta t \tag{5}$$

This is the same as $\Delta x = -\eta \nabla f \Delta t$, if the velocities are set to zero before taking each step and $\eta = \Delta t / (2\, m_k)$. Notice that the higher the masses, the lower is the learning rate, since the parameters are "harder to move". The variations of the learning rate can be seen equivalently as altering the time step: a smaller learning rate means a slower simulation, that is a smaller time step. In Sect. 3 we will study the dependence of the key hyperparameters η and $|\mathcal{B}|$ from the temperature, (see [6]), but this relation will be more elusive.

We define the *instantaneous temperature* $\mathcal{T}(t)$ of the system as the kinetic energy $\mathcal{K}(t)$ divided by the number of degrees of freedom d and a constant k_B to obtain the desired units:

$$\mathcal{T}(t) = \frac{\mathcal{K}(t)}{k_B\, d} = \frac{1}{k_B\, d} \sum_{k=1}^{d} \frac{1}{2} m_k\, v_k(t)^2$$

The *thermodynamics temperature* is then the time average of $\mathcal{T}(t)$:

$$T = \frac{1}{\tau} \int_0^\tau \mathcal{T}(t)\, dt = \frac{1}{\tau k_B\, d} \int_0^\tau \mathcal{K}(t) = \frac{K}{k_B\, d}$$

where K is the average kinetic energy and τ is long enough to yield small fluctuations in T and depends on the time scale of the individual particle motions. In practice, to perform mechanical simulations at constant (or regularly varying) temperature, coupling with a thermal reservoir is introduced by rescaling

the velocities every fixed number of steps to match the desired temperature. These are called *constant temperature* simulations; we set the temperature equal to zero every step (instead every few steps as usual). The mechanical equivalent, is a dynamic simulation where heat is extracted from the system at each step. Performing such an optimization with the gradient descent (GD), has a clear mechanical interpretation: it leads to a (local) minimum in the potential or, equivalently, a minimum of the total energy at zero temperature, when the kinetic energy vanishes.

We now turn to examine the effect of SGD, Stochastic Gradient Descent. With SGD, the gradient for a given geometry changes each time it is computed, because of the random choice of the minibatch $\mathcal{B}$. This amounts to a residual velocity associated to each particle even after equilibrium is reached. Hence, our system does not evolve according to Newton dynamics and in particular the mechanical energy is not constant.

Once the equilibrium is reached, the macroscopic parameters (the loss and temperature) will no longer change, i.e. they will have small fluctuations only. In particular $\langle v\rangle = 0$, that is, we expect the average value of velocity to vanish. Notice that here we have a key difference between GD (gradient descent) and SGD (stochastic gradient descent): at equilibrium

$$\sigma^2 = \langle v^2\rangle - \langle v\rangle^2$$

$\sigma^2 = 0$ for GD but $\sigma^2 = \langle v^2\rangle$ for SGD. This corresponds to the physical fact that GD reaches the equivalent of the zero Kelvin (no temperature), while with SGD we maintain a residual finite temperature. With a constant temperature simulation we will achieve the minimum *free energy* and not a minimum of the potential energy, that is our loss function.

We now recall the *principle of equipartition of energy*: *at thermodynamic equilibrium, all accessible degrees of freedom have, on a sufficiently long time average, the same kinetic energy.*

Let K_{av} be the time average kinetic energy of particle (i.e. parameter) k. By the principle of equipartition of energy, this is $1/d$ the total kinetic energy:

$$K_{av} = K/d = k_B T$$

Hence knowing the average value of v_k^2 at equilibrium, i.e., the variance of the gradient, this equation allows us to compute the temperature (if we set all masses equal to 1).

In the next section we will perform experiments to test our thermodynamic model and the relation between some of the notions we introduced. In Sect. 4, we shall interpret the continuous version of the time evolution of our system as the dynamics of a mechanical system with Hamiltonian H consisting of the sum of a conservative term $H^{\text{mech}} = K + V$ and a dissipative term.

3 Experiments

Our experiments were performed on both the MNIST and CIFAR10 datasets (see [9,10]) obtaining similar results. We report the experiments on the MNIST dataset only. We are using a LeNet modified architecture in colab platform (Fig. 1):

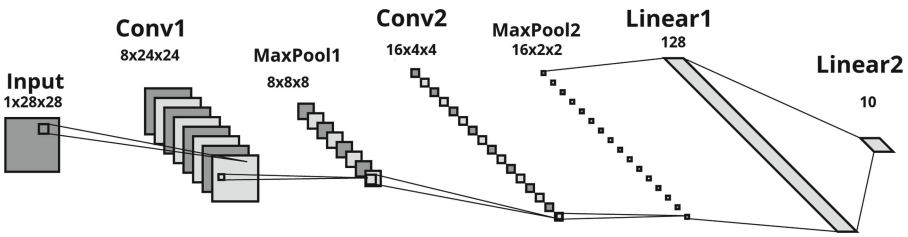

Fig. 1. Modified LeNet

This is an accurate, yet simple, network consisting of two convolutional layers (Conv1, Conv2) followed by two linear ones (Linear1, Linear2), with a low number of parameters. Batchnormalization and maxpool are also used.

We use SGD to optimize the network, with a constant for the regularization penalty $\lambda = 4 \cdot 10^{-2}$ and minibatch size $\beta = 32$. We start our training with learning rate $\eta = 10^{-2}$, then decrease it to 10^{-3} after 300 epochs and finally to 10^{-4} at 600 epochs. The loss function $f_{\mathcal{B}}$ during training and average temperature at equilibrium in layers are expressed in Fig. 2 and 3.

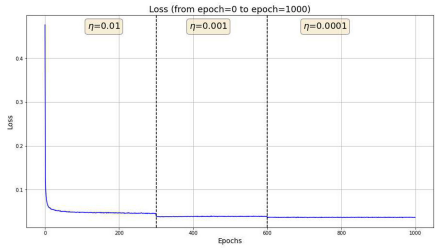

Fig. 2. Loss function

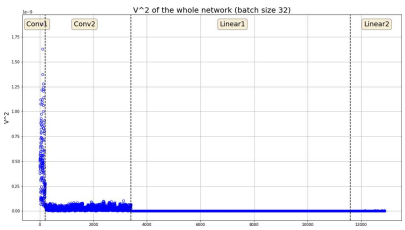

Fig. 3. Temperature in layers

Notice that different layers exhibit significantly different temperatures.

In Fig. 4 and 5 we describe the behaviour of the temperature T, as defined in our previous section, depending on the inverse $1/\beta$ of the minibatch size and the learning rate η. Despite in the literature ([6] and refs therein) T is commonly believed to behave proportionally to such parameters, we discover in practice quite a different behaviour.

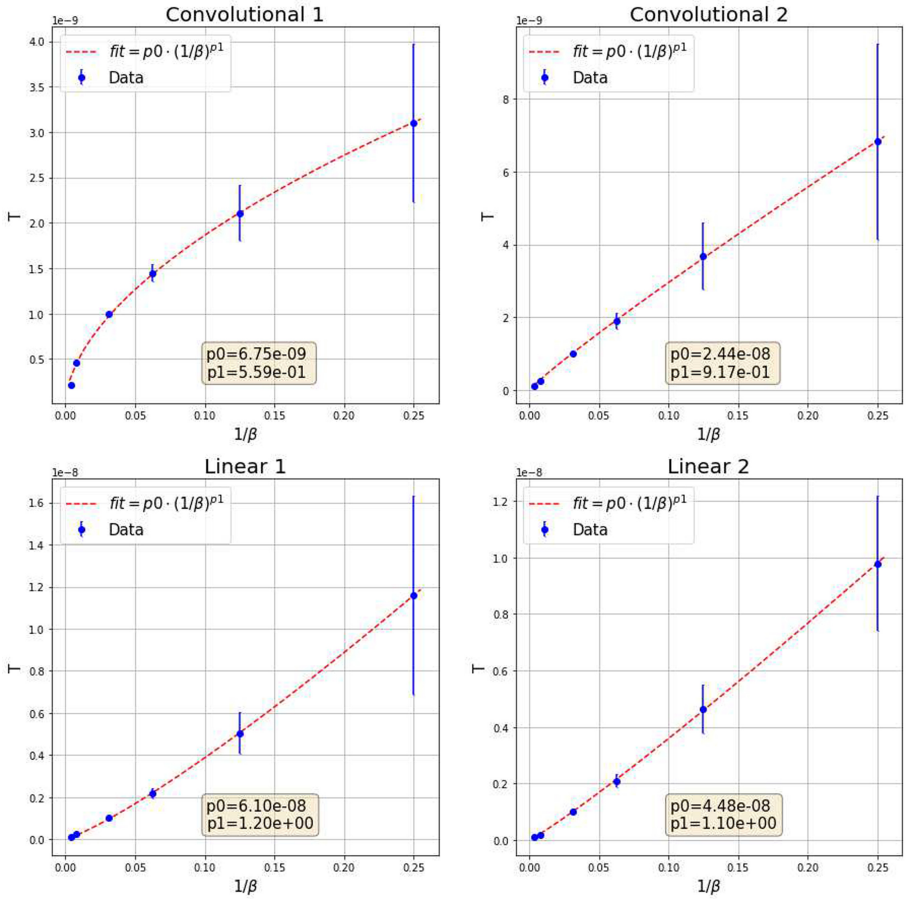

Fig. 4. Batch size and temperature

Notice first that different layers exhibit significantly different behaviour in the dependence of η and β from the temperature, hence they should be examined separately. In fact we see a linear behaviour of the temperature with respect to $1/\beta$ just for the Convolutional 2 and Linear 2. While we have an essential non linearity for the others. Similar considerations hold for the quadratic behaviour with respect to the learning rate. We now look at the temperature of the filters of the first convolutional layer at equilibrium (see the values of parameters in Conv 1 in Fig. 6).

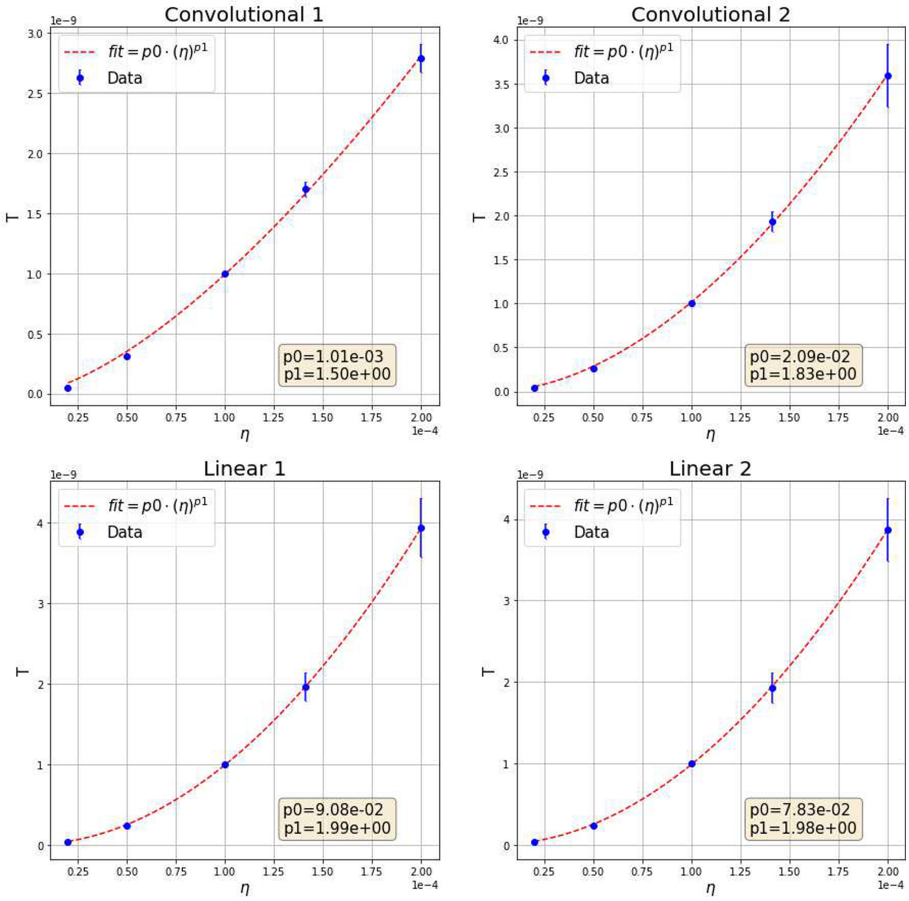

Fig. 5. Learning rate and temperature

Clearly the parameters of different filters have different temperature behaviours at equilibrium: some filters tend to stay stable while others keep changing. A possible interpretation of this is that some filters are more effective than others, so once learnt the system will not forget them. Vice versa, non effective filters in image recognition still change since they do not contribute to loss reduction.

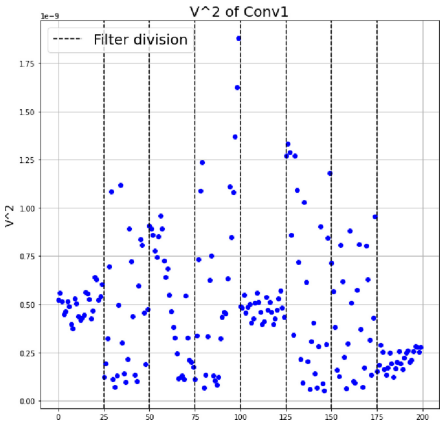

Fig. 6. Temperature of filters in the first convolutional layer

4 Contact Hamiltonian Dynamics, Lie Groups Thermodynamics and SGD

In this section we provide some mathematical insight to our thermodynamic interpretation of SGD described in Sec. 2 and the experiments in Sec. 3. Since at each step we extract heat from our system, we cannot assume that the sum of kinetic energy and potential (the loss function) H^{mech} is preserved; we need to consider a term taking into account dissipation.

We assume then (see [4], Sec. 5) that the thermodynamic space of parameters $\mathbb{R}^{d+1}$ is equipped with a contact structure:

$$\alpha = dS - p_a dq^a$$

The contact hamiltonian dynamics is then ruled by the contact hamiltonian:

$$H = H^{\text{mech}} + \mathcal{V}(S)$$

where $H^{\text{mech}} = K + V$ as above. We could take as first approximation (see [4]), $\mathcal{V}(S) = \gamma S$, where γ is a constant and S is the entropy of the system. This leads to the contact Hamilton equations:

$$\begin{cases} \dot{q}^a = \frac{\partial H}{\partial p_a} \\ \dot{p}_a = -\frac{\partial H}{\partial q^a} - p_a \frac{\partial H}{\partial S} \\ \dot{S} = p_a \frac{\partial H}{\partial p_a} \end{cases} \tag{6}$$

We plan to measure in the future, with a long enough simulation the entropy S in this context, by knowing the temperature and the area sampled by the evolution of the system in the parameter space. This will enable the modelling with a contact hamiltonian system and a concrete mean to test it. Also, the entropy comes into play in other contexts, like the Koszul-Souriau approach to thermodynamics.

We now make some considerations on Lie group thermodynamics and suggest possibly future mathematically interesting directions. Consider the action of G the Galilei group on space time. As Souriau proves, this action is hamiltonian, so it makes sense to speak of its moment map. We cannot however expect generalized Gibbs states to exist for the full Galilean group, but, as specified in [11] 7.3.3 only for one-parameter subgroups. If we are able to run our experiment long enough, the system would explore a large portion of the parameter space, so to test the partition function predicted by the probability function ([11] 7.1.1):

$$\rho_b = \frac{1}{P(b)} e^{-\langle J, b \rangle}, \qquad b \in \text{Lie}(G) \tag{7}$$

where J denotes the moment map.

We also believe that if would be mathematically interesting to fit contact dynamics for a thermodynamics system into the framework of Souriau Lie group thermodynamics and measurements on this simple model could be an experimental evidence that these these two theories are effective and equivalent ways to describe popular machine learning systems.

5 Conclusions

We described a parallelism between SGD dynamics in Deep Learning and a thermodynamics system. Experiments show that the temperatures of each layer behaves independently, hence it is necessary to treat layers as independent systems. Furthermore, the temperature of each layer does not depend in a consistent and simple way on the size of the minibatch and the learning rate: extra care must then be exerted when defining the relation of the temperature with such key hyperparameters. Insight from Lie group thermodynamics and its generalization to contact hamiltonian dynamics suggests to push this analogy further to obtain quantitative experimental results.

Acknowledgements. We are indebted with Prof. P. Chaudhari, Dr. A. Achille and Prof. S. Soatto for many helpful discussions. We also thank our Referees for valuable suggestions.

References

1. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzmann machines. Cogn. Sci. **9**(1), 147–169 (1985)
2. Amari, S.-I.: Natural gradient works efficiently in learning. Neural Comput. **10**(2), 251–276 (1998)
3. Barbaresco, F.: Lie group statistics and Lie group machine learning based on Souriau Lie groups thermodynamics and Koszul-Souriau-Fisher metric: new entropy definition as generalized Casimir invariant function in coadjoint representation. Entropy **22**(6), 642 (2020)
4. Bravetti, A.: Contact Hamiltonian dynamics: the concept and its use. Entropy **19**(10), 535 (2017)
5. Fioresi, R., Chaudhari, P., Soatto, S.: A geometric interpretation of stochastic gradient descent using diffusion metrics. Entropy **22**(1), 101 (2020)
6. Chaudhari, P., Soatto, S.: Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In: 2018 Information Theory and Applications Workshop (ITA), pp. 1–10. IEEE (2018)
7. Hopfield, J.J.: Neurons with graded response have collective computational properties like those of two-state neurons. PNAS **81**(10), 3088–3092 (1984)
8. Jaynes, E.T.: Information theory and statistical mechanics. Phys. Rev. **106**(4), 620 (1957)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.; ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105 (2012)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
11. Marle, C.-M.: From tools in symplectic and Poisson geometry to J. M. Souriau's theories of statistical mechanics and thermodynamics. Entropy **18**(10), 370 (2016)
12. Salakhutdinov, R., Hinton, G.: Deep Boltzmann machines. In: Artificial Intelligence and Statistics, pp. 448–455. PMLR (2009)
13. Simoes, A.A., De Leon, M., Valcazar, M.L. and De Diego, D.M.: Contact geometry for simple thermodynamical systems with friction. Proc. Roy. Soc. A **476**(2241), 20200244 (2020)