Partial local entropy and anisotropy in deep weight spaces

Daniele Musso[*]

Departamento de Física de Partículas, Universidade de Santiago de Compostela (USC), Instituto Galego de Física de Altas Enerxías (IGFAE), E-15782 Santiago de Compostela, Spain;
Inovalabs Digital S.L. (TECHEYE), E-36202 Vigo, Spain;
and Centro de Supercomputación de Galicia (CESGA), s/n, Avenida de Vigo, 15705, Santiago de Compostela, Spain

(Received 25 September 2020; revised 13 February 2021; accepted 9 March 2021; published 5 April 2021)

We refine a recently proposed class of local entropic loss functions by restricting the smoothening regularization to only a subset of weights. The new loss functions are referred to as *partial* local entropies. They can adapt to the weight-space anisotropy, thus outperforming their isotropic counterparts. We support the theoretical analysis with experiments on image classification tasks performed with multilayer, fully connected, and convolutional neural networks. The present study suggests how to better exploit the anisotropic nature of deep landscapes, and it provides direct probes of the shape of the minima encountered by stochastic gradient descent algorithms. As a byproduct, we observe an asymptotic dynamical regime at late training times where the temperature of all the layers obeys a common cooling behavior.

DOI: 10.1103/PhysRevE.103.042303

I. INTRODUCTION

Recent studies on the weight space of deep neural networks [1,2] have highlighted the existence of rare subdominant clusters of configurations that yield a high test accuracy. Although these clusters constitute a deviation from typicality, they are efficiently encountered by *stochastic gradient descent* (SGD) algorithms and correspond to wide valleys of suitable loss functions, such as cross entropy [3].

An analogous circumstance occurs in the context of constraint satisfaction problems, where the chase after clusters of solutions is improved when the loss function gets supplemented by a term that encourages a local high density of solutions [4]. To find the number of solutions contained in the vicinity of a specific weight configuration, one can define a local solution-counting functional, namely a *local entropy*.

Classification tasks performed by means of quantized neural networks (where the weights are discrete) can be interpreted as constraint satisfaction problems. There are, however, two reasons to generalize the concept of local entropy: First, classification problems are typically required to reach a high but not necessarily perfect accuracy; second, they are often approached with machines that have continuous weights.[1] The strict counting of solutions employed for constraint satisfaction problems can therefore be relaxed to just an incentive that encourages a high local density of high-accuracy configurations. A local averaging of the loss, for instance, is expected to have such an effect, but other deformations of the loss yielding a local smoothening can be valid choices, too.

A specific smoothening procedure of the loss function can be enforced by means of a spatial convolution with a Euclidean heat kernel, whose spread is controlled by a parameter γ,

$$\mathcal{F}(\beta,\gamma;\boldsymbol{W}) = -\ln \int d^N W' \exp\Big(-\beta \mathcal{L}(\boldsymbol{W}') - \frac{\gamma}{2}||\boldsymbol{W}-\boldsymbol{W}'||_2^2 \Big), \tag{1}$$

where both $\boldsymbol{W}$ and $\boldsymbol{W}'$ parametrize the N-dimensional weight space, $||\cdot||_2$ represents the Euclidean norm, and $\mathcal{L}$ is a generic loss function; adopting an energetic interpretation of the loss, the parameter β corresponds to an inverse temperature. In the limit $\beta \to 0$, the integral in (1) can be interpreted as (the continuum version of) a weighted counting of the configurations $\boldsymbol{W}'$ where the weighting decreases exponentially with their distance from $\boldsymbol{W}$ [5].

The smoothening introduced by (1) is *isotropic* in weight space. However, when optimizing with SGD, the gradient noise depends in general on both the position and the direction, this being actually a key factor for the success of SGD algorithms [6]. Therefore, it is natural to expect that a refinement of the smoothening functional able to suitably exploit the anisotropy of gradient noise can significantly improve its regularizing effects. Besides, such refinement can furnish an interesting new probe of the weight space.

The present paper focuses on *partial*, entropic and local smoothening, namely a smoothening analogous to (1) applied to just a subsets of weights. This allows one to address weight-space anisotropy in a direct and active way. We will loosely adopt the term *partial local entropy* to convey this idea irrespective of the details of the particular smoothening technique, as long as it corresponds to an incentive to local high density of high-accuracy configurations restricted to a subset of weights.[2]

[*]daniele.musso@usc.es; daniele.musso@cesga.es; mudaniele@yahoo.com

[1]Up to the numerical precision employed.

[2]The functional $\mathcal{F}(\beta,\gamma;\boldsymbol{W})$ defined in (1) can be interpreted in analogy to a thermodynamical potential; as such, it should be referred

II. ANISOTROPY IN WEIGHT SPACE

By definition, the neurons of a deep network are arranged on different layers, and such architecture imposes a natural hierarchy among them, according to their depth within the network. In a fully connected setting, the receptive field of each neuron coincides with the whole input, however deeper neurons are fed with signals that have been preprocessed by lower-lying neurons. Roughly, while the neurons in the first layer compute a weighted sum of the network inputs, the neurons in the second layer compute a weighted sum of the outputs of the first layer, that is, a weighted sum of a weighted sum of the network inputs. Such compositional nature of the operation performed by each subsequent layer suggests that the depth of the network corresponds to a hierarchy in combinatorial complexity [7].[3] Any isotropic assumption about the weight space neglects this structural hierarchy, thereby it should be regarded with caution, if not suspicion.

Careful consideration of the hierarchical anisotropy of the weight space has led to important insight about the inner workings of neural networks (also in the biological domain [8]) as well as improvements in the optimization of artificial neural networks.[4] Gradient noise depends on both position and direction, and its covariance matrix is correlated to the Hessian matrix of the loss function, which makes SGD escape exponentially fast from sharp minima [6].[5] Thus, it is fair to consider weight-space anisotropy as one of the main features at the root of the effectiveness of SGD algorithms in reaching high test accuracy and generalization.

Layer temperature and asymptotic cooling

The learning dynamics of a deep neural network trained with SGD is in general a complex process. The system is out of equilibrium and, given the dependence of the gradient noise on the position in weight space, one cannot schematize the training as the evolution of a system in contact with an equilibrium thermal reservoir. Nonetheless, it is still possible to define a temperature as the variance of the gradient noise when schematizing the training evolution in terms of a Brownian motion [7,11,12]. More precisely, one has to focus on the covariance matrix $D(\boldsymbol{W})$ characterizing the stochastic Wiener process.[6]

Let us focus for a moment on a specific point $\boldsymbol{W}^*$ in weight space. Given the anisotropy of $D(\boldsymbol{W}^*)$, it is impossible to define a unique temperature characterizing all directions, but one can in principle still define a temperature for each direction. Since we are working in a space with very high dimensionality, this is hardly of any help. However, we should recall that there is a natural grouping of the directions in weight space provided by the layered structure of the network. Furthermore, it is possible to define layer variables that average over the weights belonging to the same layer. One can consider fluctuations of such layer variables that, due to the averaging over a layer, are expected to be stabler and reflect the hierarchy of the architecture. Accordingly, one can define a layer temperature corresponding to the variance of such layer averaging of gradients.[7] This corresponds to regarding the layers as if they were the individual units of a neural network; despite being a crude approximation, this could help us to gain useful insight about the training dynamics [14].[8]

The layer temperature is a characterization of the noise of the training signal s_I through layer I, defined as

$$s_I = \frac{1}{N_I} \sum_{\omega \in \Omega_I} ||\nabla_\omega \mathcal{L}(\boldsymbol{W})||_2, \tag{2}$$

where Ω_I denotes the set of the N_I weights connecting the Ith layer with its inputs, $||\cdot||_2$ represents the Euclidean norm, and $\mathcal{L}(\boldsymbol{W})$ is the loss evaluated at the weight configuration $\boldsymbol{W}$. The training signals s_I and their noise evolve during optimization, and it is possible to isolate different regimes in the training dynamics. In [14] the authors observed that a possibly generic dynamic transition occurs when the signal-to-noise ratio switches from being initially dominated by the signal to being later dominated by noise. This occurs quite abruptly (in terms of optimization time) and approximately at the moment when the training signal attains its maximum value; see Fig. 1.

The numerical studies that we performed suggest the generic presence of a further dynamic transition, occurring at later stages of the training. This eventual regime is characterized by a subexponential decay of both signal and noise for all layers. Interestingly, the subexponential contraction of the signal and the noise for all the layers is characterized by a common decaying behavior. At late times, the hierarchy between layers is therefore preserved and gets frozen: the

to as local *free* entropy; this extra connotation is sometime omitted to avoid clutter.

[3]One can rephrase such combinatorial complexity in terms of correlations among the input channels: the neurons in the first layer are sensitive to the inputs individually, so they respond to one-point correlations; the neurons belonging to the nth layer, instead, are sensitive to n-point correlations, that is, the joint correlations of n inputs.

[4]In this regard, two relevant examples are Kaiming weight initialization [9] and regularization by means of anisotropic noise injection [10,11].

[5]To keep the analysis as simple as possible, in the present paper we do not exploit the Hessian matrix of the loss function to define specific partial local entropies, yet this represents an interesting direction for further investigation. Specifically, information about the eigenvalues of the Hessian matrix could be useful in *scoping* the hyperparameter γ [see (1)], namely in adjusting its value during optimization in an adaptive fashion.

[6]We refer to [12] for the definition of the covariance matrix $D(\boldsymbol{W})$. The analysis of a Brownian motion by means of the Fokker-Planck equation encodes both the noise anisotropy and its dependence on position through the covariance matrix $D(\boldsymbol{W})$ [12,13].

[7]We underline that a direct analysis of the variance of the gradient noise for the single weights shows that in general the weights belonging to the same layer *cannot* be characterized by a common temperature. Said otherwise, the possibility of defining a layer temperature does not imply thermal isotropy within the subspace spanned by the weights of the same layer.

[8]Reference [14] has been debated in the subsequent literature, where there is a wider set of references useful for a critical analysis [15,16].

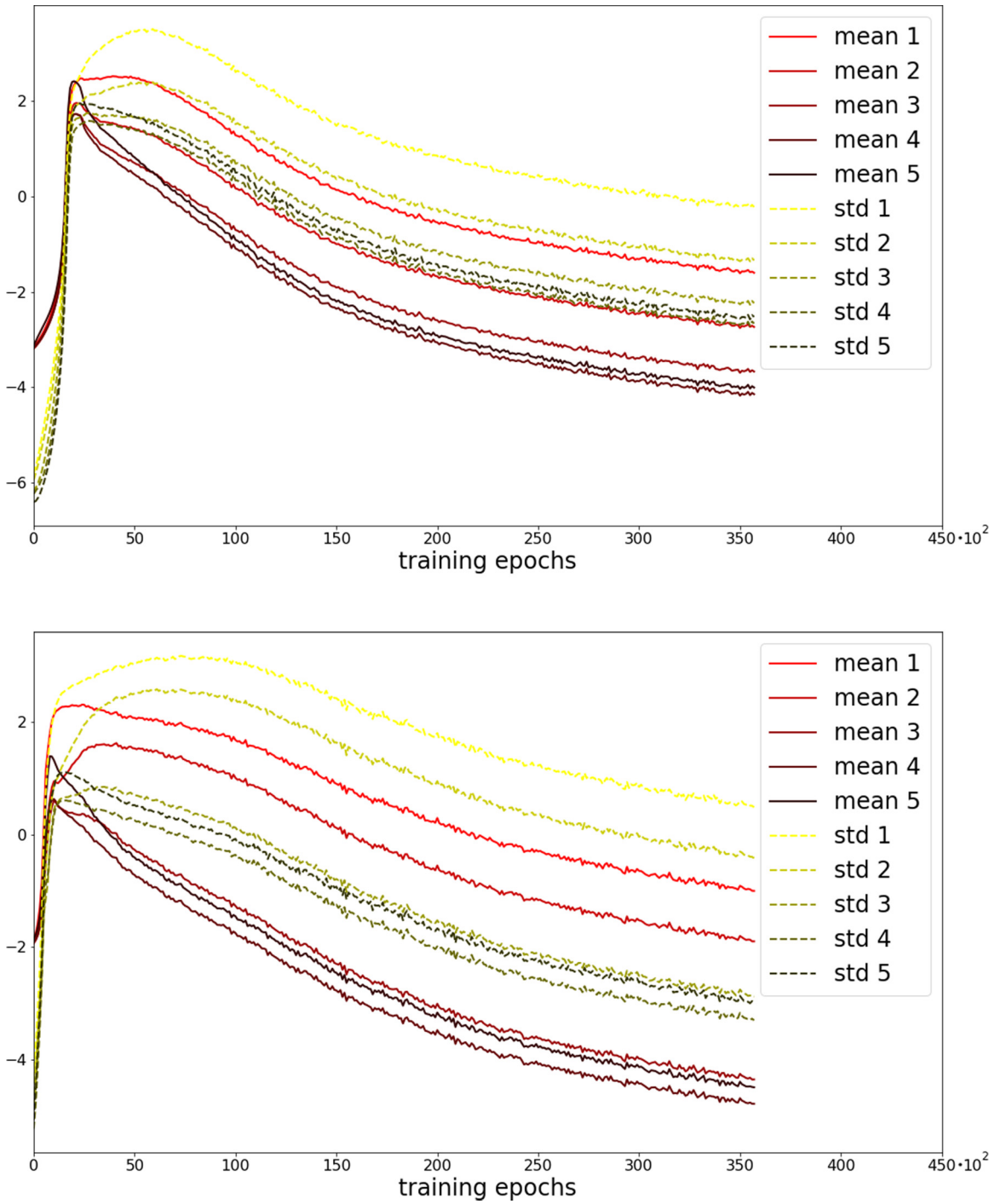

FIG. 1. The training signal s_I defined in (2)—where $I \in \{1, 2, 3, 4\}$ labels the layers of the network—is represented with solid lines; the dashed lines represent the associated standard deviations. The plots depict a long training of two four-layer fully connected neural networks on MNIST with either ReLU (top plot) or TanH (bottom plot) activation functions. Three distinct dynamical regimes emerge in both plots: (i) an early signal-dominated regime; (ii) a short, intermediate, and noise-dominated regime; and (iii) an eventual, long, and noise-dominated regime where all quantities decay subexponentially with a common behavior (the vertical axes are natural logarithms).

dynamics of all the layers can in fact be described factorizing the common subexponential decay.

Interpreting the noise as a temperature and adopting a renormalization group language, the eventual subexponential cooling (possibly turning exponential at asymptotically late times) is suggestive of an infrared fix point, where quantities evolve by a common rescaling without distortion at asymptotic low energies.[9] It is relevant to stress that Fig. 1 has been obtained *without* adopting weight-decay regularization. Moreover, we have obtained qualitatively similar results both

[9]Here, a potential connection emerges to studies of neural networks under the perspective of scaling rules; see, for instance, [17,18].

with ReLU and TanH activation functions; while the former is scale covariant, the latter is not.

As already stressed, even if the layerwise account gives a very coarse-grained picture of the actual training dynamics, still it confirms the importance of anisotropy throughout the whole training process, including at asymptotic late times where the in-sample loss and the test error have long stabilized.

III. PARTIAL LOCAL FREE ENTROPY

For the sake of generality, the present section is rather technical. The reader who is just interested in the specific losses used in the experiments can jump to Sec. IV and focus on the loss functions (22) and (23) without missing the core ideas.

We consider the cross-entropy loss $\mathcal{L}_{\text{c.e.}}(\boldsymbol{W})$ as the baseline function to be smoothened; $\boldsymbol{W}$ is a vector indicating a configuration in weight space. We consider y additional configurations $\boldsymbol{W}+\Delta\boldsymbol{W}^a$ with $a=1,\dots,y$, shifted by a uniformly distributed random vector $\Delta\boldsymbol{W}^a$. The loss corresponding to each configuration is supplemented by an additional term measuring its distance from the unperturbed point $\boldsymbol{W}$. For the moment we let the distance function $d_{R,k}(\Delta\boldsymbol{W}^a)$ be arbitrary, but we assume that it depends on two parameters, to be specified later. We consider the new loss

$$\mathcal{M}(R,k,y;\boldsymbol{W}) \equiv -\ln\left\{\frac{1}{y+1}\left[e^{-\mathcal{L}_{\text{c.e.}}(\boldsymbol{W})} + \sum_{a=1}^{y} e^{-\mathcal{L}_{\text{c.e.}}(\boldsymbol{W}+\Delta\boldsymbol{W}^a)-d_{R,k}(\Delta\boldsymbol{W}^a)}\right]\right\}, \tag{3}$$

normalized with respect to the number of sampling points $y+1$. Roughly, the loss $\mathcal{M}$ amounts to the logarithm of an average of exponentials. In the case of just one sampling point, $y=0$, $\mathcal{M}$ coincides with the baseline loss,

$$\mathcal{M}(R,k,y=0;\boldsymbol{W}) = \mathcal{L}_{\text{c.e.}}(\boldsymbol{W}). \tag{4}$$

We choose the following distance function:

$$d_{R,k}(\Delta\boldsymbol{W}) \equiv -\ln\prod_{i=1}^{N}\left[\left(1-\frac{1}{1+e^{-2k(\Delta W_i-R)}}\right)\times\frac{1}{1+e^{-2k(\Delta W_i+R)}}\right], \tag{5}$$

which depends on two real parameters, R and k. In the $k\to\infty$ limit, the kernel

$$K_{R,k}(\Delta\boldsymbol{W}) \equiv e^{-d_{R,k}(\Delta\boldsymbol{W})} \tag{6}$$

reduces to the characteristic function of the N-dimensional hypercube $H_{\boldsymbol{W},R}$ centered in $\boldsymbol{W}$ with edges $2R$ long,[10]

$$\lim_{k\to+\infty} K_{R,k}(\Delta\boldsymbol{W}) = \prod_{i=1}^{N}[1-\Theta(\Delta W_i-R)]\Theta(\Delta W_i+R). \tag{8}$$

Thus, the parameter R represents the effective linear size of the support of the kernel (6), while k controls its sharpness; see Fig. 2. In the infinite sharpness limit, $k\to\infty$, the random displacement vectors $\Delta\boldsymbol{W}^a$ in (3) are sampling the hypercube $H_{\boldsymbol{W},R}$ uniformly.

Taking an infinite number of sampling points,

$$\mathcal{M}(R,k,y;\boldsymbol{W}) \xrightarrow[y\to+\infty]{} \mathcal{F}(R,k;\boldsymbol{W}), \tag{9}$$

where

$$\mathcal{F}(R,k;\boldsymbol{W}) \equiv -\ln\int d^N W'\, e^{-\mathcal{L}_{\text{c.e.}}(\boldsymbol{W}')}\, K_{R,k}(\boldsymbol{W}'-\boldsymbol{W}), \tag{10}$$

defines a parametric family $\mathcal{F}(R,k;\boldsymbol{W})$ of *local free entropies*, in analogy with (1).[11] Taking the $k\to\infty$ limit of (10), one obtains

$$\lim_{k\to+\infty}\mathcal{F}(R,k;\boldsymbol{W}) = -\ln\int_{H_{\boldsymbol{W},R}} d^N W'\, e^{-\mathcal{L}_{\text{c.e.}}(\boldsymbol{W}')}. \tag{12}$$

To recapitulate, in the limit of a large number of sampling points, $y\to\infty$, the loss function $\mathcal{M}(R,k,y;\boldsymbol{W})$ approximates a parametric family of *free local entropy* functions (10) parametrized by the effective linear size R of the smoothening region (in weight space) and the sharpness k of the associated kernel (6).

To define *partial local free entropies*, we have only to generalize the passages above to the case in which only a subset of weights is smoothened over. We can define a discrete indicator function $\boldsymbol{U}$ taking values in $\{0,1\}^N$ and defined on the N dimensions of weight space: it takes value 1 on the directions along which we smoothen the loss, and 0 on the remaining directions in weight space. Thinking of $\boldsymbol{U}$ as an N-dimensional vector, it provides an un-normalized projector onto the subset of weights considered for smoothening. We can thus define a restricted version of the distance function $d_{R,k}(\Delta\boldsymbol{W})$,

$$d^{[\boldsymbol{U}]}_{R,k}(\Delta\boldsymbol{W}) \equiv d_{R,k}((\Delta\boldsymbol{W}\cdot\boldsymbol{U})\boldsymbol{U}), \tag{13}$$

where $\cdot$ indicates the scalar product of $\mathbb{R}^N$ in the N-dimensional weight space.

---

[10] Recall that the Heaviside step function $\Theta(x)$ can be obtained as the limit of infinite sharpness for a sigmoid function, namely

$$\Theta(x) = \lim_{k\to+\infty}\frac{1}{1+e^{-2kx}}. \tag{7}$$

[11] The particular local free entropy specified in (1) is associated with a different choice of distance, namely

$$d(\gamma;\Delta\boldsymbol{W}) = \gamma||\Delta\boldsymbol{W}||_2^2. \tag{11}$$

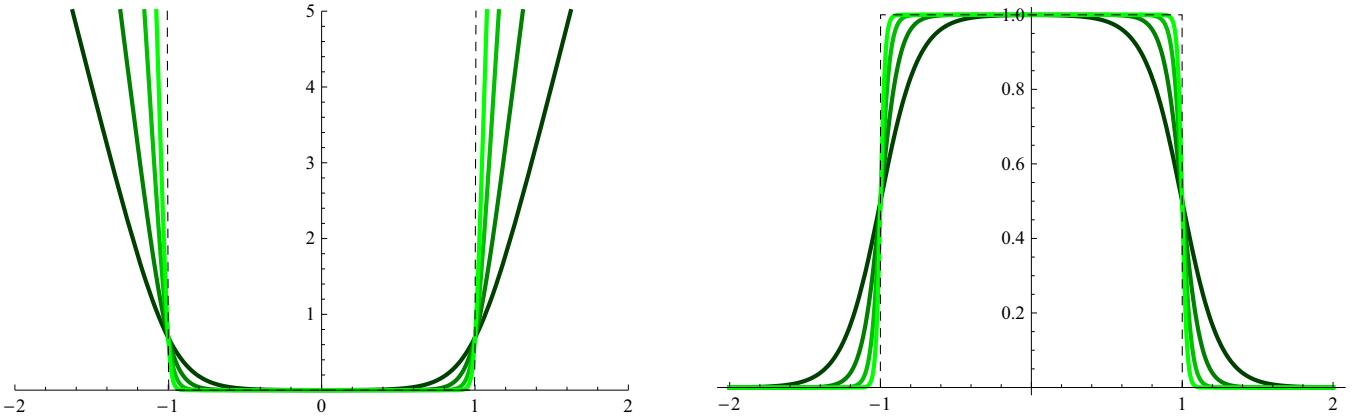

FIG. 2. One-dimensional section of the distance function $d_{R,k}$ defined in (5) (left plot) and of the kernel $K_{R,k}$ defined in (6) (right plot); in the plots $R = 1$ and $k = 2^2, 2^3, 2^4, 2^5$ from darker to lighter.

Adopting the restricted distance (13), we can repeat the same steps as above: first consider

$$\mathcal{M}^{[U]}(R,k,y;\boldsymbol{W}) \equiv -\ln\left\{\frac{1}{y+1}\left[e^{-\mathcal{L}_{\text{c.e.}}(\boldsymbol{W})} + \sum_{a=1}^{y} e^{-\mathcal{L}_{\text{c.e.}}(\boldsymbol{W}+\Delta\boldsymbol{W}^a)-d^{[U]}_{R,k}(\Delta\boldsymbol{W}^a)}\right]\right\}, \quad (14)$$

then take the $y \to \infty$ limit

$$\mathcal{M}^{[U]}(R,k,y;\boldsymbol{W}) \xrightarrow[y\to+\infty]{} \mathcal{F}^{[U]}(R,k;\boldsymbol{W}), \quad (15)$$

where

$$\mathcal{F}^{[U]}(R,k;\boldsymbol{W}) \equiv -\ln\int d^N W'\, e^{-\mathcal{L}_{\text{c.e.}}(\boldsymbol{W}')}\, K^{[U]}_{R,k}(\boldsymbol{W}'-\boldsymbol{W}) \quad (16)$$

represents a parametric family of *partial local free entropies*. Eventually, take the $k \to \infty$ limit,

$$\mathcal{F}^{[U]}(R;\boldsymbol{W}) \equiv \lim_{k\to+\infty} \mathcal{F}^{[U]}(R,k;\boldsymbol{W}), \quad (17)$$

where

$$\mathcal{F}^{[U]}(R;\boldsymbol{W}) \equiv -\ln\int_{H^{[U]}_{\boldsymbol{W},R}} d^N W'\, e^{-\mathcal{L}_{\text{c.e.}}(\boldsymbol{W}')}; \quad (18)$$

the integration region $H^{[U]}_{\boldsymbol{W},R}$ is a hypercube extended only in the directions along which $\boldsymbol{U}$ is non-null.

A simpler entropic loss

It is interesting to seek a simpler loss that could somehow preserve the smoothening effect of partial local free entropy. To this end, one can define an averaged loss over an N-dimensional vicinity in weight space—this imitating the effects of local entropy—or to a lower-dimensional vicinity—this instead imitating partial local entropy. We focus on the latter case and define

$$\bar{\mathcal{L}}^{[U]}(R,k,y;\boldsymbol{W}) \equiv \frac{1}{y+1}\left[\mathcal{L}_{\text{c.e.}}(\boldsymbol{W}) + \sum_{a=1}^{y} \mathcal{L}_{\text{c.e.}}(\boldsymbol{W}+\Delta\boldsymbol{W}^a) K^{[U]}_{R,k}(\Delta\boldsymbol{W}^a)\right]. \quad (19)$$

Considering the $k \to \infty$ limit, one obtains

$$\bar{\mathcal{L}}^{[U]}(R,y;\boldsymbol{W}) \equiv \frac{1}{y+1}\left[\mathcal{L}_{\text{c.e.}}(\boldsymbol{W}) + \sum_{a=1}^{y} \mathcal{L}_{\text{c.e.}}(\boldsymbol{W}+\Delta^{[U]}\boldsymbol{W}^a)\right], \quad (20)$$

where $\Delta^{[U]}$ means simply that the random vectors are sampled uniformly within the hypercube $H^{[U]}_{\boldsymbol{W},R}$ centered in $\boldsymbol{W}$ and extending along the direction indicated by the vector $\boldsymbol{U}$, its edges being $2R$ long. In the limit of infinite samples, we have

$$\bar{\mathcal{L}}^{[U]}(R;\boldsymbol{W}) \xrightarrow[y\to+\infty]{} \int_{H^{[U]}_{\boldsymbol{W},R}} d^N W'\, \mathcal{L}_{\text{c.e.}}(\boldsymbol{W}'), \quad (21)$$

and the loss reduces to a simple local average along a subset of directions in weight space.[12]

IV. EXPERIMENTS WITH FULLY CONNECTED NETWORKS

The focus of the first group of experiments is on layerwise partial entropy regularizations for multilayer, fully connected neural networks trained on image classification tasks. Namely, we considered partial local entropies where the subset of weights chosen for smoothening coincides with whole layers. We consider the 10-class classification tasks associated with MNIST [20] and FASHION-MNIST [21] datasets, whose input images are 28 pixels wide and 28 pixels high. We consider both two- and three-layer fully connected neural networks with continuous weights[13] having a further 10-neuron output layer. All layers except the last have $784 = 28^2$ neurons and are structurally identical, apart from their different depth within the network. The following hyperparameters have been kept fixed for all the experiments: learning rate $\eta = 0.0001$, momentum $\mu = 0.9$, minibatch size 256, and trained for 120 epochs.

[12]The loss function defined in (19) can be related to the *robust ensemble* studied in [5], which in turn is similar to the *elastic averaging* proposed in [19].

[13]We performed the experiments with single floating point numerical precision.

We considered two loss functions, a partial local exponential average loss (PLEA)

$$\mathcal{L}_{\mathrm{PLEA}}(\boldsymbol{W}) = -\ln\left\{\frac{1}{1+y}\left[e^{-\mathcal{L}_{\mathrm{c.e.}}(\boldsymbol{W})}+\sum_{a=1}^{y} e^{-\mathcal{L}_{\mathrm{c.e.}}(\boldsymbol{W}+\Delta\boldsymbol{W}^a)}\right]\right\}, \tag{22}$$

and a partial local average loss (PLA)

$$\mathcal{L}_{\mathrm{PLA}}(\boldsymbol{W}) = \frac{1}{1+y}\left[\mathcal{L}_{\mathrm{c.e.}}(\boldsymbol{W}) + \sum_{a=1}^{y} \mathcal{L}_{\mathrm{c.e.}}(\boldsymbol{W}+\Delta\boldsymbol{W}^a)\right], \tag{23}$$

where $\mathcal{L}_{\mathrm{c.e.}}$ is the cross-entropy loss and $\Delta\boldsymbol{W}^a$ is a random vector sampled in the vicinity of $\boldsymbol{W}$.[14] Such a vicinity is a hypercube centered in $\boldsymbol{W}$ with edge $2R$ and extending only along a subspace of the N-dimensional weight space. Notice that in this way the regularizations of the cross-entropy $\mathcal{L}_{\mathrm{c.e.}}$ given by (22) and (23) enforce an anisotropic bias.

In the experiments reported below, we consider only subspaces associated with one or more layers at a time.[15] Apart from the entropic smoothening, we do not enforce any further regularization; in particular, we do not use weight decay.

A. Results

The experiments suggest two main conclusions:

(i) In general, the entropic regularizations (22) and (23) improve test accuracy. The effect increases rapidly with the size R of the smoothening region, up to a maximum size beyond which performance gets degraded.

(ii) When implemented on suitable subsets of weights (e.g., single layers), the entropic regularizations outperform significantly their isotropic counterparts.

The first point means that smoothening improves performance up to a point beyond which its averaging effect distorts the original loss landscape too heavily. The second point means that the strong differences in the role played by the various weights affect the loss landscape and the effectiveness of regularization. This implies that the shape of the wide flat minima encountered by SGD optimization is relevant, not only their extension. Another generic conclusion suggested by the experiments is that the layerwise entropic regularization is more effective when performed on deeper levels. This harmonizes with the intuitive idea that deeper weights are associated with more complex features, which, in a reliable classification, should be progressively more robust.

An important detail of the experimental setups is that all layers have the same number of neurons, 784. Thus, when comparing quantities associated with different layers, we are actually probing the mere effect of depth. A direct comparison between structurally different layers would instead be more difficult to interpret.

[14]The losses (22) and (23) correspond to infinite sharpness limits, $k \to \infty$, of (14) and (19), respectively. See Sec. III for more details.

[15]Throughout the present paper, the weight space spanned by $\boldsymbol{W}$ is formed only by the synaptic coefficients connecting different layers, while it excludes biases. Despite these biases being present and trained over, we do not smooth over them.

B. Two-layer fully connected neural network on FASHION-MNIST

We considered two-layer, fully connected neural networks adopting both PLEA loss function (22) and PLA loss function (23). The results obtained with the two loss functions are qualitatively analogous.

We measured the test accuracy reached by three versions of the same two-layer network as we moved the regularization radius R.[16] The three versions differ simply by the choice of the weight subspace considered for smoothening: either (i) the whole first layer; (ii) the whole second layer; or (iii) both layers (isotropic choice). The results are reported in Figs. 3 and 4 (left plot). Regularization on the second layer alone proved to be the best strategy for both choices of loss functions and in the entire range of R probed by the experiments. The isotropic regularization can outperform the regularization on the first layer alone, but only at very small values for R. In fact, the isotropic choice leads soon to degraded results as R increases, while the single-layer regularizations continue to improve the test accuracy, showing a saturating behavior.

C. Three-layer fully connected neural network on MNIST

The experiments on the three-layer fully connected neural networks confirm and extend the results obtained for its two-layer counterpart. They are depicted in Fig. 4 (right plot). In particular, the isotropic choice proves to be the worst among all the possible choices of subsets[17] as soon as the smoothening radius R is sufficiently big. Moreover, there is an articulated interplay of regimes as R varies: at the lowest values of R, the best choice consists in regularizing with respect to the first and third layers jointly; at large values of R, regularizing with respect to the second or third layer alone proves to be the best choice. Also, the performance hierarchy among the suboptimal regularization schemes changes as R moves, showing a complicated structure.

D. Finer sampling

To test whether the decrease in accuracy associated with regularizing on multiple layers is due to insufficient sampling [i.e., too small y; see (22) and (23)], we repeated the experiments performed with the two-layer fully connected neural network on FASHION-MNIST with PLA loss doubling the number of sampling points y. The results obtained with $y = 8$ are comparable to those obtained with $y = 4$; see Fig. 5; this hints at the fact that the sampling of the smoothening neighborhood cannot explain the poor performance of multilayer regularization.

V. EXPERIMENTS WITH CONVOLUTIONAL NEURAL NETWORKS

We extend the analysis described in Sec. IV in two directions: (i) we consider more complicated image classification

[16]That is, the parameter encoding the linear size of the smoothening region; see Sec. III for details.

[17]Recall that we consider only subsets of weights associated with one or more whole layers.

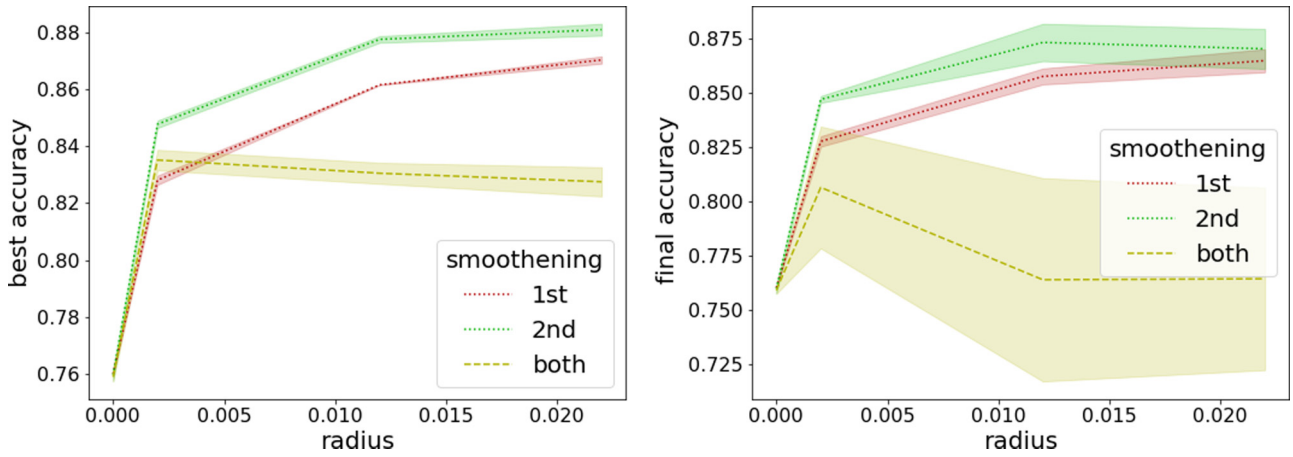

FIG. 3. Comparison of the best (left) and final (right) test accuracy reached by a two-layer fully connected neural network on FASHION-MNIST. The lines correspond to three different PLA losses [see (23)] obtained by smoothening the cross entropy, respectively, on the first, the second, or both layers.

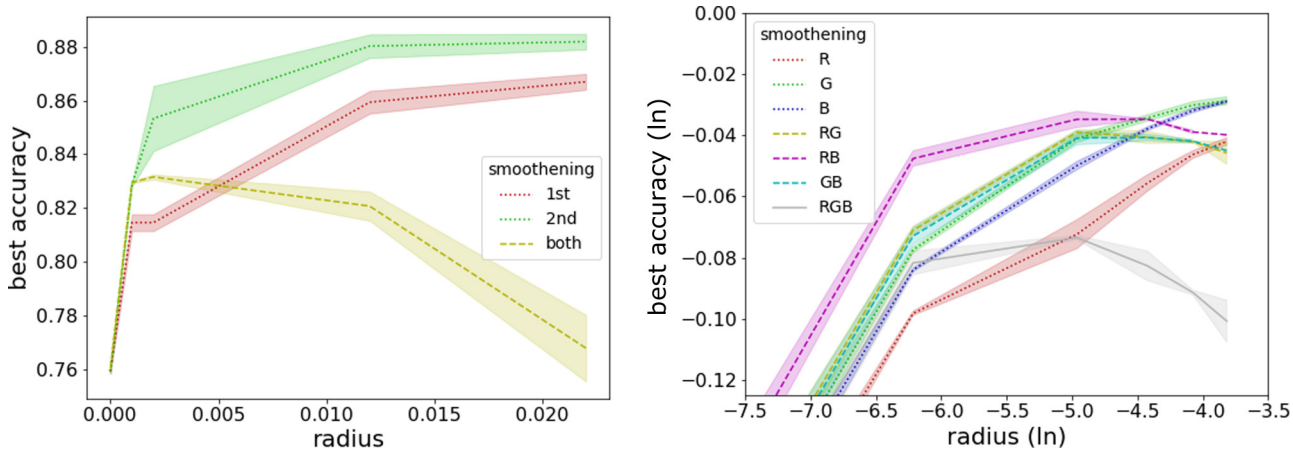

FIG. 4. Left plot: best test accuracy reached during training by a two-layer fully connected neural network over FASHION-MNIST. The three lines correspond to three different PLEA regularization schemes [see Eq. (3)], where smoothening is performed on the first layer alone, on the second layer alone, or on both layers, respectively. Right plot: best test accuracy reached by a three-layer fully connected neural network on MNIST. The lines represents different PLA regularization schemes according to an RGB color nomenclature, where red corresponds to the first layer, green to the second, and blue to the third.

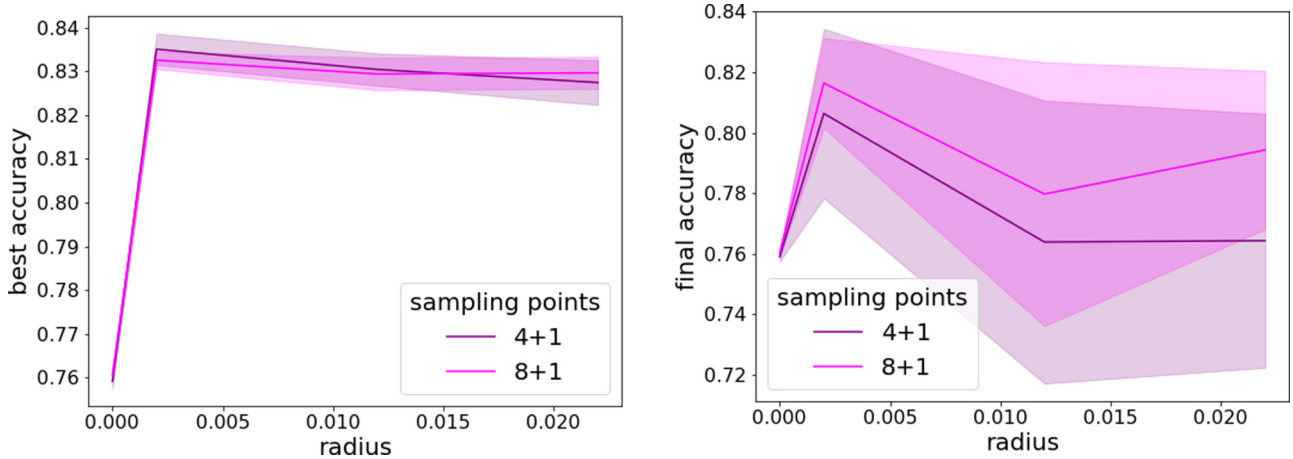

FIG. 5. Comparison of the test accuracy performance obtained with a bilayer fully connected neural network on FASHION-MNIST and trained with PLA loss [see Eq. (23)]. The lighter line refers to finer sampling, $y = 8$, while the darker line refers to $y = 4$. There is no strong sensitivity to the sample size.

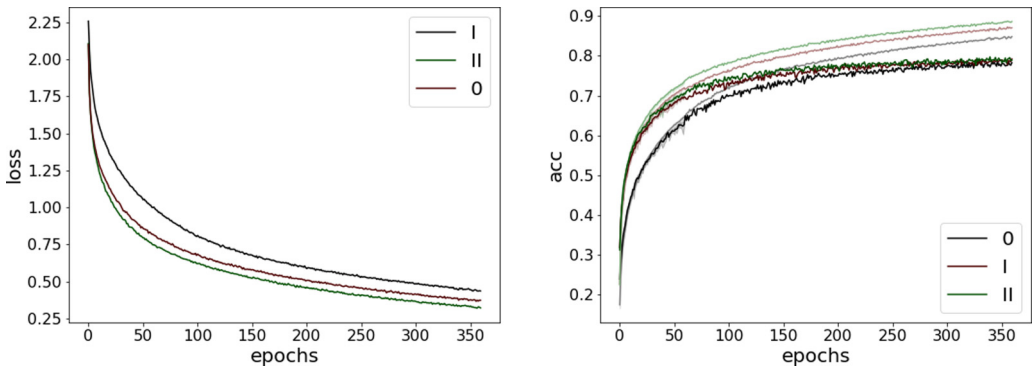

FIG. 6. Training loss (left) and test accuracy (right) of the convolutional network (24) trained to solve the CIFAR10 image classification task. The black lines refer to the case in which no entropic regularization was considered. The red and green lines refer to PLA [see (23)] loss applied to the first (I) or second (II) fully connected layer, respectively, as specified in the legend. The paler lines in the right plot depict the in-sample accuracy.

tasks; and (ii) we consider deeper networks with convolutional architectures. The convolutional neural networks that we adopt present five convolutional layers followed by three fully connected layers; the details of the architectures are given in (24) and (25). All the convolutional kernels are 3×3.

A. Results

The main conclusions that emerged from the study of partial entropic regularizations applied to convolutional neural networks are the following:

(i) Partial entropic regularizations involving the convolutional layers lead, in general, to worse classification accuracy with respect to the nonregularized case.

(ii) When applied to the fully connected head of convolutional networks, partial entropic regularizations can improve the classification accuracy in a similar way to that observed on multilayered perceptrons, described in Sec. IV, especially when adopting an *early stopping* strategy interrupting training before its full stabilization.

Some further comments are in order. Convolutional layers implement a structured bias encoding some degree of *locality* and *translational invariance*. Thus the convolutional structure, if compared to fully connected layers, is highly specialized. Entropic regularizations can in general be thought of as corresponding to the integration over some injected artificial noise. As such, one expects them to weaken, if not to spoil, any specific bias previously encoded in the neural architecture. Such comment holds both for the partial entropic regularization studied here as well as for other forms of noisy regularizations, like Dropout. The latter, too, has been observed to hamper the performance of convolutional networks [22]. Conversely, fully connected layers have no specific structure, and the average over additional noise can lead to better performance in general, also when applied to the fully connected head in a convolutional network.

In the experiments detailed below, we consider fully connected heads formed by three layers. The deepest layer outputs 10 channel, as required by the 10-class classification tasks considered, and we do not regularize it. The other two fully connected layers are instead equal in shape among themselves. As already argued in Sec. IV for the multilayer perceptrons, the structural equality allows for a direct comparison between the two layers.

B. CIFAR10

For the classification task corresponding to the CIFAR10 dataset [23], we considered the following convolutional architecture:

Layer	In channels	Put channels
Conv	3	64
Conv	64	64
MaxPool		
Conv	64	128
Conv	128	128
Conv	128	128
MaxPool		
Fully	$128 \times 4 \times 4$	$128 \times 4 \times 4$
Fully	$128 \times 4 \times 4$	$128 \times 4 \times 4$
Fully	$128 \times 4 \times 4$	10

(24)

We have trained it for 360 epochs with a constant learning rate $\eta = 10^{-4}$, a minibatch size of 256 images, and momentum $\mu = 0.9$ without Nesterov acceleration. The training dataset has been augmented and regulated by means of random transformations on the images. Specifically, we have considered rescaled random crops ranging from 60% to 100% of the image area and with a height-to-width ratio from $\frac{3}{4}$ to $\frac{4}{3}$. Neither weight decay nor dropout layers have been used.[18] Actually, the only regularization for the stochastic gradient descent has been provided by the partial local average, encoded in (23), with four additional sample points drawn from a uniform distribution in a hypercube ball of a side equal to $2R$, with $R = 0.01$. The initialization followed the so-called Kaiming procedure described in [24].

The results are depicted in Fig. 6. We considered three cases: no regularization or PLA regularization applied to either the first or second layer in the fully connected head [see

[18] We compare the partial entropic regularizations against weight-decay regularizations in Sec. V B 1.

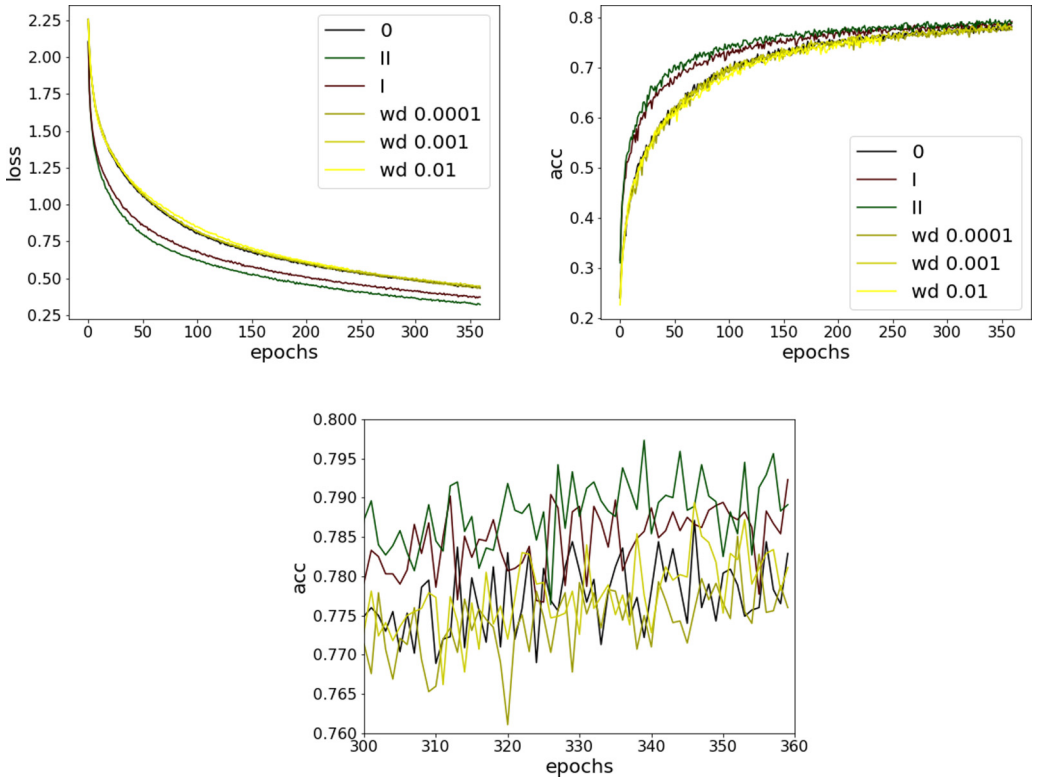

FIG. 7. The paler (yellow) lines represent training instances that adopted increasing levels of weight-decay regularization, 0.01, 0.001 and 0.0001, respectively, from darker to lighter. They all overlap with the unregularized case (label 0), meaning that the effects of weight regularization are irrelevant. The plot on the top left depicts the in-sample loss during training on CIFAR10; the lowest line corresponds to II and the second from below corresponds to I. The plot on the top right shows the out-of-sample accuracy during training on CIFAR10; the uppermost line corresponds to II while the second from the top corresponds to I. The bottom plot is a zoom of the top right figure highlighting the late portion of the training.

(24)]. The PLA modification of the loss function yields better performance, both in-sample and out-of-sample, especially if combined with an early stopping strategy, which interrupts the training before its eventual stabilization. The PLA procedure implies collecting multiple samples of the loss function in the vicinity of the current weight configuration of the network (we took four points in a hypercubic vicinity plus the center); the gradient is accumulated but eventually rescaled in such a way that the multiple sampling does not affect the training by means of a simple amplification of the learning rate.

Comparison against standard weight-decay regularization

To better assess the effects of partial entropic regularization, we considered comparing them with those produced by a standard regularization method, namely weight decay [25]. Specifically, we considered three levels of weight-decay rate: 0.01, 0.001, and 0.0001. As shown in Fig. 7, weight-decay regularization proved to be of essentially no use in the present experiments. On the contrary, partial entropic regularization improved the performance, more significantly in the early phase of the training but only slightly in later stages. These experiments do not pretend to support a generic claim; however, they show explicitly that partial entropic regularization can be preferable with respect to weight regularization.

C. STL10

STL10 is a 10-class classification dataset [26] of 96×96 color images acquired from ImageNet [27]. STL10 was designed for partially unsupervised learning [28]. In fact, it contains only 500 labeled images for supervised training. Although these hardly suffice to train a machine in a fully supervised setup, we use them to simply show the positive effects that partial entropy regularizations induce on the early phase of the training, without requiring an overall satisfactory performance.[19]

We adopt the following convolutional architecture:

Layer	In channels	Out channels
Conv	3	8
Conv	8	8
MaxPool		
Conv	8	16
Conv	16	16
Conv	16	16
MaxPool		
Fully	$16 \times 20 \times 20$	$16 \times 20 \times 20$
Fully	$16 \times 20 \times 20$	$16 \times 20 \times 20$
Fully	$16 \times 20 \times 20$	10

(25)

[19]STL10 has already been used in the literature for supervised learning; see, for example, [29].

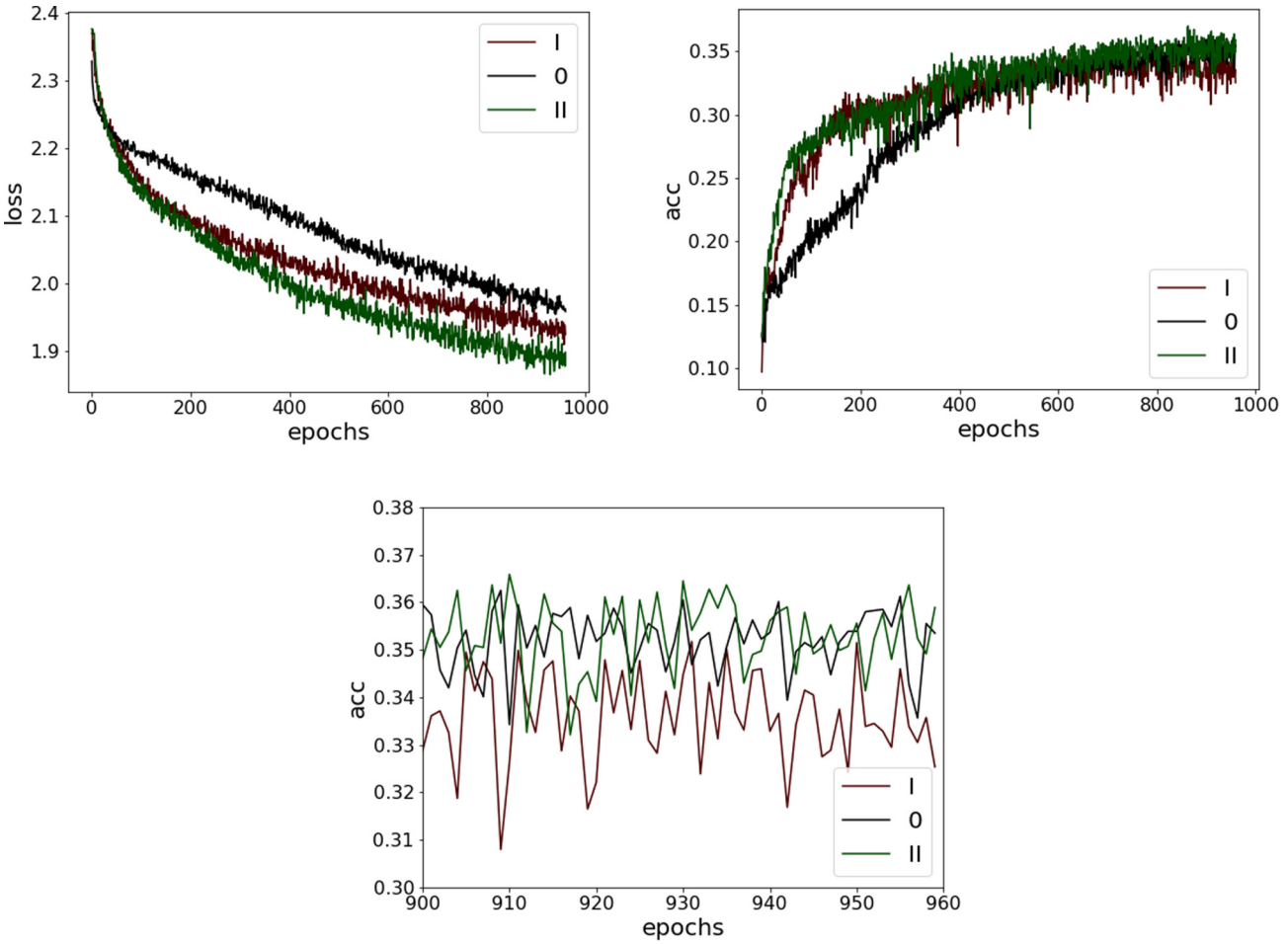

FIG. 8. Training loss (left) and test accuracy (right) of a deep convolutional network (25) trained on the STL10 image dataset in a fully supervised scheme for 960 epochs. The black lines correspond to zero entropic regularization, while the lines I and II correspond to partial entropic regularization applied only to the first and second fully connected layers, respectively. The green line (II) corresponds to lowest loss and highest accuracy. The bottom plot is a zoom over the last part of the right plot above, depicting the accuracy levels reached at the end of the training.

which is analogous to (24) but has lighter layers. We trained it for 960 epochs with a constant learning rate $\eta = 10^{-5}$, momentum $\mu = 0.9$ without Nesterov acceleration, and a minibatch size of 64 images. To mitigate the issue presented by the smallness of the training set, we have applied heavy augmentation and regularization to the training images. Specifically, we considered random crops whose size ranges from 8% to the full image, and whose aspect ratio ranges from $\frac{3}{4}$ to $\frac{4}{3}$; we considered random horizontal flips, random reduction to gray-scale (with a probability $p = 0.1$), color jitter (brightness, contrast, saturation, and hue all set to 0.5), and random rotation whose maximal rotation angle is $\pm\pi$ radians.[20]

We monitored the training, and we report the evolution of the in-sample loss and the test accuracy in Fig. 8. The partial entropic regularization, applied to one layer at a time, improves the training and validation performances, but only if accompanied by an early stopping strategy. The experiments of Fig. 8 refer to a PLA loss (23) where the side of the sampled hypercube is $2R$ with $R = 0.01$. The network was initialized according to the Kaiming method [24]; no weight decay was considered.

VI. DISCUSSION

A local smoothening of the loss function can improve the chase for wide flat minima [1,2], which is already a strength of the standard stochastic gradient descent algorithm [6].[21] We elaborate and refine the smoothening techniques based on local entropy to the purpose of leveraging the anisotropic nature of deep weight spaces. Concretely, we propose to restrict local entropic losses to suitable sub-spaces of weights, thus defining *partial local entropies*. This allows us to explore, address, and exploit the intrinsic anisotropic nature of deep weight spaces. In fact, we show that a partial entropic regularization can

[20]To implement such transformations, we relied on the *transforms* library in PyTorch [30].

[21]The generic relevance of *wide flat minima* is still debated in the literature [31,32], especially in relation to scale covariance and normalization in weight space for networks adopting ReLU activations.

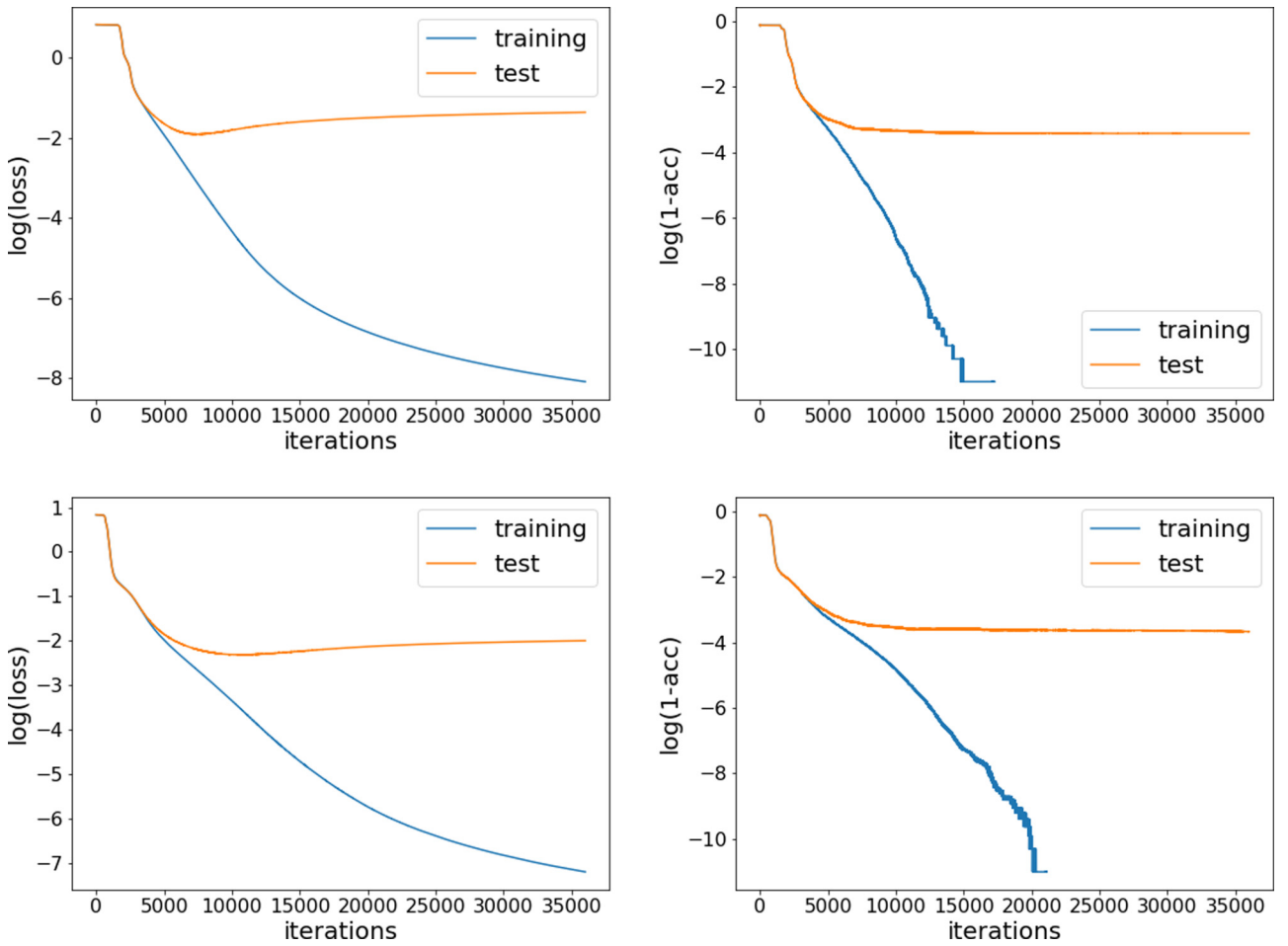

FIG. 9. Loss and accuracy levels during training, both in-sample and out-of sample, for the five-layer perceptron (A1) with ReLU (top) and TanH (bottom) activation functions. Logarithms are in the natural basis.

implement useful biases on the shape of the minima encountered by SGD optimization.

We have mainly explored the layerwise implementations of partial local entropies; although there is room for finer analyses resolving smaller subspaces, the layerwise approach is both natural (i.e., well-adapted to the architecture of deep networks) and informative.

In the present paper, we have applied partial entropic regularizations to some fully connected and convolutional neural networks employed for image classification tasks. They can, however, be employed for the optimization of wider classes of learning machines, e.g., autoencoders [33]. In particular, the specific layerwise entropic regularizations proposed in the present study apply in any context involving a layered neural network. The partial entropic regularizations have been proved to be potentially useful in all the considered experiments. However, their positive effects in progressively more demanding tasks seem to be restricted to an early stopping protocol. The adoption of a partial entropic loss led to a more aggressive optimization in all the performed experiments.

Direct analysis in the language of statistical physics

The study of local entropic regularizations is a very active research front in machine learning, especially in connection to statistical physics [1,3–5,19,33–35]. Wide flat minima have been described as a structural characteristic of deep networks, and their correlation with good generalization performance has been claimed in [1,3]. In some simple setups, it is even possible to estimate analytically the hypervolume of the clusters of configurations giving rise to the relevant minima [3,36]. The theoretical framework on which the calculations are based has been developed for the study of disordered systems in condensed matter, mainly spin glasses (see [37] and references therein), and it is called the *replica approach*. Within this approach, different regimes are described by different *Ansätzes*, and they can be separated by clustering transitions [38].[22]

A simple version of the replica approach [41] can rely on two (crude) assumptions: (i) averaging over (typically Gaussian) input; (ii) considering treelike architectures. The former essentially washes out completely the information about the dataset. This is not always undesirable; in fact, it allows for the characterization of structural properties of the machines that hold true *per se* independently of the dataset. However, it constitutes a limitation whenever the actual information

[22] An analogous transition in K-SAT problems has been studied in [39,40].

provided by the input is important. As a future prospect, it would be interesting to study how a direct and explicit account of correlations in the input data could improve the theoretical understanding of the partial entropic regularizations, especially regarding their effects on the inference quality.[23] Considering a treelike architecture is very helpful to simplify the computations; in fact, avoiding loops in the network often opens the possibility of exact computations by, for instance, belief propagation algorithms [3,41]. Nevertheless, adopting a treelike network as a proxy for a fully connected one can be too crude a simplification, which is expected to deviate more significantly as the depth of the system is increased.

To explain the experiments described in Sec. IV, it would be desirable to have a direct control on the shape of the relevant clusters of weight configurations reached upon SGD training, or at least an estimation thereof. This could be seen as a refinement of the estimation of the clusters size [3,36], and as such it is likely to be a very demanding endeavor up to the point that it becomes natural to ask whether some simpler—though possibly rougher—approach is viable. To this end, it is interesting to investigate mean-field inference methods [43].

ACKNOWLEDGMENTS

I would like to thank Riccardo Argurio, Diogo Buarque Franzosi, Stefano Goria, Javier Más, Andrea Mezzalira, Giorgio Musso, Alfonso Ramallo, Hernán Serrano, and Maurice Weiler for interesting discussions.

[23]The study performed in [42] is relevant to this purpose, nevertheless it would require a generalization beyond single-layer networks.

APPENDIX: DETAILS ON THE ASYMPTOTIC COOLING EXPERIMENTS

The experiments described in Sec. II A were performed with a five-layer, fully connected neural network with the following architecture:

Layer	In channels	Out channels
Fully connected	28^2	28^2
Fully connected	28^2	28^2
Fully connected	28^2	28^2
Fully connected	28^2	28^2
Fully connected	28^2	10

(A1)

The following hyperparameters have been adopted: learning rate $\eta = 10^{-4}$, momentum $\mu = 0.9$ without Nesterov acceleration, minibatch size of 256 images. The networks have been trained for 3.5×10^4 epochs on the MNIST dataset. Neither weight decay nor partial local entropic regularizations have been used. The only difference between the two experiments relies in the activation functions, ReLU and TanH, respectively. The loss and accuracy levels during training are reported in Fig. 9.

[1] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, Subdominant Dense Clusters Allow for Simple Learning and High Computational Performance in Neural Networks with Discrete Synapses, Phys. Rev. Lett. **115**, 128101 (2015).

[2] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina, Entropy-SGD: Biasing gradient descent into wide valleys, arXiv:1611.01838.

[3] C. Baldassi, E. M. Malatesta, and R. Zecchina, Properties of the Geometry of Solutions and Capacity of Multilayer Neural Networks with Rectified Linear Unit Activations, Phys. Rev. Lett. **123**, 170602 (2019).

[4] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, Local entropy as a measure for sampling solutions in constraint satisfaction problems, J. Stat. Mech.: Theor. Expt. (2016) 023301.

[5] C. Baldassi, C. Borgs, J. T. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes, Proc. Natl. Acad. Sci. (USA) **113**, E7655 (2016).

[6] Z. Xie, I. Sato, and M. Sugiyama, A diffusion theory for deep learning dynamics: Stochastic gradient descent escapes from sharp minima exponentially fast, arXiv:2002.03495.

[7] W. E, A proposal on machine learning via dynamical systems, Commun. Math. Stat. **5**, 1 (2017).

[8] M. Riesenhuber and T. Poggio, Hierarchical models of object recognition in cortex, Nat. Neurosci. **2**, 1019 (1999).

[9] K. He, X. Zhang, S. Ren, and J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, CoRR, arXiv:1502.01852.

[10] Z. Zhu, J. Wu, B. Yu, L. Wu, and J. Ma, The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects, arXiv:1803.00195.

[11] D. Musso, Stochastic gradient descent with random learning rate, arXiv:2003.06926.

[12] P. Chaudhari and S. Soatto, Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks, CoRR, arXiv:1710.11029.

[13] P. da Silva, L. da Silva, E. Lenzi, R. Mendes, and L. Malacarne, Anomalous diffusion and anisotropic nonlinear fokker-planck equation, Physica A **342**, 16 (2004).

[14] R. Shwartz-Ziv and N. Tishby, Opening the black box of deep neural networks via information, CoRR, arXiv:1703.00810.

[15] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox, On the information bottleneck theory of deep learning, in *International Conference on Learning Representations* (ICLR, 2018).

[16] Z. Goldfeld, E. van den Berg, K. Greenewald, I. Melnyk, N. Nguyen, B. Kingsbury, and Y. Polyanskiy, Estimating information flow in deep neural networks, arXiv:1810.05728.

[17] S. L. Smith and Q. V. Le, A Bayesian perspective on generalization and stochastic gradient descent, arXiv:1710.06451.

[18] U. Sharma and J. Kaplan, A neural scaling law from the dimension of the data manifold, arXiv:2004.10802.
[19] S. Zhang, A. Choromanska, and Y. LeCun, Deep learning with Elastic Averaging SGD, arXiv:1412.6651.
[20] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE **86**, 2278 (1998).
[21] H. Xiao, K. Rasul, and R. Vollgraf, FASHION-MNIST: A novel image dataset for benchmarking machine learning algorithms (2017).
[22] T. DeVries and G. W. Taylor, Improved Regularization of Convolutional Neural Networks with Cutout, arXiv:1708.04552.
[23] https://www.cs.toronto.edu/~kriz/cifar.html.
[24] K. He, X. Zhang, S. Ren, and J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, arXiv:1502.01852.
[25] A. Krogh and J. A. Hertz, A simple weight decay can improve generalization, in *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91 (Morgan Kaufmann, San Francisco, CA, USA, 1991), pp. 950–957.
[26] https://cs.stanford.edu/~acoates/stl10/.
[27] http://www.image-net.org/.
[28] A. Coates, H. Lee, and A. Ng, An analysis of single-layer networks in unsupervised feature learning, **1**, 01 (2011) .
[29] M. Weiler and G. Cesa, General $E(2)$-Equivariant Steerable CNNs, arXiv:1911.08251.
[30] https://pytorch.org/docs/stable/torchvision/transforms.html.
[31] Q. Liao, B. Miranda, L. Rosasco, A. Banburski, R. Liang, J. Hidary, and T. Poggio, Generalization puzzles in deep networks, in *International Conference on Learning Representations* (ICLR, 2020).
[32] T. Poggio, A. Banburski, and Q. Liao, Theoretical issues in deep networks, Proc. Natl. Acad. Sci. (USA) **117**, 30039 (2020).
[33] M. Negri, D. Bergamini, C. Baldassi, R. Zecchina, and C. Feinauer, Natural representation of composite data with replicated autoencoders, arXiv:1909.13327.
[34] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina, Entropy-SGD: Biasing Gradient Descent Into Wide Valleys, arXiv:1611.01838.
[35] C. Baldassi, R. D. Vecchia, C. Lucibello, and R. Zecchina, Clustering of solutions in the symmetric binary perceptron, arXiv:1911.06756.
[36] E. Barkai, D. Hansel, and I. Kanter, Statistical Mechanics of a Multilayered Neural Network, Phys. Rev. Lett. **65**, 2312 (1990).
[37] M. Mezard, G. Parisi, and M. Virasoro, *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*, World Scientific Lecture Notes in Physics (World Scientific, Singapore, 1987.
[38] T. Castellani and A. Cavagna, Spin-glass theory for pedestrians, J. Stat. Mech.: Theory Exp. (2005) P05012.
[39] M. Mézard, G. Parisi, and R. Zecchina, Analytic and algorithmic solution of random satisfiability problems, Science **297**, 812 (2002).
[40] F. Krzakała, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová, Gibbs states and the set of solutions of random constraint satisfaction problems, Proc. Natl. Acad. Sci. (USA) **104**, 10318 (2007).
[41] M. Mézard and A. Montanari, *Information, Physics, and Computation*, Oxford Graduate Texts (Oxford University Press, Oxford, 2009).
[42] Y. Kabashima, Inference from correlated patterns: a unified theory for perceptron learning and linear vector channels, J. Phys.: Conf. Ser. **95**, 012001 (2008).
[43] M. Gabrié, Mean-field inference methods for neural networks, J. Phys. A **53**, 223002 (2020).