A jamming transition from under- to over-parametrization affects loss landscape and generalization

Stefano Spigler [1]
stefano.spigler@epfl.ch

Mario Geiger [1]
mario.geiger@epfl.ch

Stéphane d'Ascoli [2]
stephane.dascoli@ens.fr

Levent Sagun [1]
levent.sagun@epfl.ch

Giulio Biroli [2]
giulio.biroli@ens.fr

Matthieu Wyart [1]
matthieu.wyart@epfl.ch

[1] Institute of Physics, École Polytechnique Fédérale de Lausanne
1015 Lausanne, Switzerland

[2] Laboratoire de Physique Statistique, École Normale Supérieure, PSL Research University
75005 Paris, France

Abstract

We argue that in fully-connected networks a phase transition delimits the over- and under-parametrized regimes where fitting can or cannot be achieved. Under some general conditions, we show that this transition is sharp for the hinge loss. In the whole over-parametrized regime, poor minima of the loss are not encountered during training since the number of constraints to satisfy is too small to hamper minimization. Our findings support a link between this transition and the generalization properties of the network: as we increase the number of parameters of a given model, starting from an under-parametrized network, we observe that the generalization error displays three phases: *(i)* initial decay, *(ii)* increase until the transition point — where it displays a cusp — and *(iii)* power law decay toward a constant for the rest of the over-parametrized regime. Thereby we identify the region where the classical phenomenon of over-fitting takes place, and the region where the model keeps improving, in line with previous empirical observations for modern neural networks. The theoretical results presented here appeared in [18] for a physics audience. The results on generalization are new.

1 Introduction

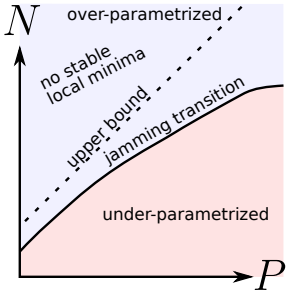
Despite the remarkable progress in designing [27, 19] and training [21] neural networks, there is still no general theory explaining their success, and their understanding remains mostly empirical. Central questions need to be clarified, such as what conditions need to be met in order to fit data properly, why the dynamics does not get stuck in spurious local minima, and how the depth of the network affects its loss landscape.

Complex physical systems with non-convex energy landscapes featuring an exponentially large number of local minima are called glassy [5]. An analogy between deep networks and glasses has been proposed [10, 8], in which the learning dynamics is expected to slow down and to get stuck in the highest minima of the loss. Yet, several numerical and rigorous works [16, 43, 20, 40, 9] suggest a different landscape geometry where the loss function is characterized by a connected level set. Furthermore, studies of the Hessian of the loss function [37, 38, 3] and of the learning

dynamics [29, 2] support that the landscape is characterized by an abundance of flat directions, even near its bottom, at odds with traditional glassy systems.

In the last decade physicists have unveiled an analogy between the physical phenomenon of jamming [22], which characterizes the onset of rigidity in disordered packings of particles, and phase transitions taking place in certain classes of computational optimization and learning problems [26, 45, 15], in particular the perceptron [14, 15] (see also [13]). In this work we push this analogy further and show that deep fully-connected networks undergo a jamming transition: above it they reach zero loss and do not get stuck in a bad minima, whereas below it they get stuck at a finite value of the loss, both for real data (images) and random data. When the hinge loss is used, the transition is sharp[1].

Furthermore, we observe that generalization properties are strongly affected by the proximity to the jamming transition. In particular, the generalization error displays a cusp[2] at the critical point [1, 28]. The observed increase in the error until the transition point is reminiscent of the classical over-fitting phenomena that can be tackled either by imposing conditions on the weights or by early stopping [7, 33, 41, 25, 1]. Further in the over-parametrized phase, the generalization error decreases monotonically which is also in line with previous observations for modern neural networks [32, 31, 4]. We argue that the decrease in accuracy follows a power law $\epsilon_0 + \epsilon_1 \left(N/N^*\right)^{-\alpha}$ at $\alpha \approx \frac{1}{4}$, where N^* is the critical number of parameters. Thereby, we propose a complete characterization of different phases in the (N, P) space, and suggest further directions for future research.



N : #degrees of freedom
P : #training examples

2 Theoretical framework

We consider a binary classification problem, with a set of P distinct training data denoted $\{(\mathbf{x}_\mu, y_\mu)\}_{\mu=1}^P$. The vector $\mathbf{x}_\mu$ is the input, which lives in dimension d, and $y_\mu = \pm 1$ is its label. We denote by $f(\mathbf{x}; \mathbf{W})$ the output of a network corresponding to an input $\mathbf{x}$, parametrized by $\mathbf{W}$ (including both weights and biases). The parameters are learned by minimizing the quadratic hinge loss:

$$\mathcal{L}(\mathbf{W}) = \frac{1}{P} \sum_{\mu=1}^{P} \frac{1}{2} \max\left(0, \Delta_\mu\right)^2 \equiv \frac{1}{P} \sum_{\mu \in m} \frac{1}{2} \Delta_\mu^2, \tag{1}$$

where $\Delta_\mu \equiv 1 - y_\mu f(\mathbf{x}_\mu; \mathbf{W})$ and m is the set of patterns with $\Delta_\mu > 0$ and contains N_Δ elements. These patterns describe *unsatisfied constraints*: they are either incorrectly classified or classified with an insufficient margin (whereas patterns with $\Delta_\mu < 0$ are learned with margin 1). We are interested in the transition between an over-parametrized phase where the network can satisfy all the constraints ($\mathcal{L} = 0$) and an under-parametrized phase where some constraints remain unsatisfied ($\mathcal{L} > 0$).

In our approach, N will be the effective number of parameters, defined as the dimension of the subspace of $\mathbf{W}$ that has an impact on $\mathcal{L}$. It can be smaller than the number of parameters, for example due to symmetries present in the network (e.g. the scale symmetry in ReLU networks reduce one degrees of freedom per node). In Appendix A we discuss this point in detail. In our experiments, we observe that N is very close to the total number of parameters.

When the ratio $r = P/N$ is lowered starting from a large value, the optimal loss approaches zero ($\mathcal{L} \to 0$) and $\Delta_\mu \to 0 \; \forall \mu \in m$, and at the jamming transition, $\mathcal{L}$ becomes exactly zero. As argued in [42], for each $\mu \in m$ the constraint $\Delta_\mu \approx 0$ defines a manifold of dimension $N - 1$[3]. Satisfying N_Δ

[1] As a benchmark, we retrained the implementation in [17] by replacing the cross entropy by the hinge loss (adapted for multiple classes) without any other modifications. Using three different seeds, we obtained 3.61%, 3.65% and 3.82% errors, which compare well to the 3.68% errors reported in the article. See `https://github.com/mariogeiger/pytorch_shake_shake` for the source code.

[2] As a complementary point, a cusp in generalization has been found in several perceptron problems when the ratio of number of training examples by the number of parameters is tuned [36, 12, 6].

[3] Related arguments were recently made for a quadratic loss [9]. In that case, we expect the landscape to be related to that of floppy spring networks, whose spectra are predicted in [11].

such equations thus generically leads to a manifold of solutions of dimension $N - N_\Delta$[4]. Imposing that a solution exists implies that $N_\Delta \leq N^*$ where N^* is the value of N at jamming.

An opposite bound can be obtained by considerations of stability as for spheres [44], by imposing that in a stable minimum the Hessian must be positive definite. The Hessian matrix follows:

$$\mathcal{H}_\mathcal{L} = \frac{1}{P}\sum_{\mu\in m} \nabla\Delta_\mu \otimes \nabla\Delta_\mu + \frac{1}{P}\sum_{\mu\in m} \Delta_\mu \nabla\otimes\nabla\Delta_\mu \equiv \mathcal{H}_0 + \mathcal{H}_p. \tag{2}$$

$\mathcal{H}_0$ is positive semi-definite: it is the sum of N_Δ rank-one matrices, thus $\mathrm{rk}(\mathcal{H}_0) \leq N_\Delta$, implying that the kernel of $\mathcal{H}_0$ is at least of dimension $N - N_\Delta$. $\mathcal{H}_p$ in general is neither positive nor negative definite. Let us denote by E_- the negative eigenspace[5] of $\mathcal{H}_p$, and call N_- its dimension very close to jamming. Stability then imposes that $\ker(\mathcal{H}_0) \cap E_- = \{0\}$, which is only possible if $N_\Delta \geq N_-$, which gives:

$$r_c \equiv \frac{P}{N^*} \geq \frac{N_\Delta}{N^*} \geq \frac{N_-}{N^*} \equiv C_0. \tag{3}$$

We shall assume that the fraction C_0 of negative eigenvalues of $\mathcal{H}_p$ does not vanish in the large-N limit. In Appendix B we argue that in the case of ReLU activation functions and random data the spectrum of $\mathcal{H}_p$ is symmetric and $C_0 = 1/2$ independently of depth. Yet, with the ReLU, $f(\mathbf{x}; \mathbf{W})$ is not continuous and presents cusps, so that Eq. (3) needs to be modified. Introducing the number of directions N_c presenting cusps, stability implies $N_\Delta > N_- - N_c$ and $r_c \geq 1/2 - N_c/N^*$. Empirically we find that $N_c/N^* \in (0.21, 0.25)$ both for random data and images as reported in Appendix C, implying $r_c \geq 0.25$.

Our analysis supports that (i) in the case of hinge loss there is a sharp transition for for $N^* \leq C_0 P$, below which (under-parametrized phase) the loss converges to some non-zero value and above which (over-parametrized phase) it becomes null; (ii) at that point the fraction N_Δ/N of unsatisfied constraints per degree of freedom jumps to a finite value. In the two next sections we confirm these predictions in numerical experiments and observe the generalization properties at and beyond the transition point[6].

Note that the present analysis is also informative on the behavior of the spectrum of the Hessian near the transition, as argued and confirmed empirically in [18].

3 Location of the jamming transition

Here we present the numerical results on random data (uniformly distributed on a hyper-sphere and with random labels $y_\mu = \pm 1$) and on the MNIST dataset (partitioned into two groups according to the parity of the digits and with labels $y_\mu = \pm 1$). In order not to have most of the weights in the first layer, we reduce the actual input size by retaining only the first $d = 10$ principal components that carry the most variance (this hardly diminishes the performance). Further description of the protocols is in Appendix D.

In Fig. 1A,C we show the location of boundary N^* versus the number of samples P. Varying input dimension, depth and loss function (cross entropy or hinge) has little effect on the transition. This result indicates that in the present setup the ability of fully-connected networks to fit random data does not depend crucially on depth. Fig. 1C shows also a comparison of random data with MNIST. From the analysis of Section 2, the number of constraints per parameter N_Δ/N is expected to jump discontinuously at the transition (Fig. 1B,D). A difference between random data and images is that the minimum number of parameters N^* needed to fit the real data is significantly smaller and grows less fast as P increases — for $P \gg 1$, $N^*(P)$ could be sub-linear or even tend to a finite asymptote: how the data structure affects $N^*(P)$ is an important questions for future studies.

[4]Note that this argument implicitly assumes that the N_Δ constraints are independent. In disordered systems this assumption is generally correct, but it may break down if symmetries are present. See Appendix A.

[5]The negative eigenspace is the subspace spanned by the eigenvectors associated with negative eigenvalues.

[6]The source code used to generate the simulations in this paper is available at `https://github.com/mariogeiger/nn_jamming`.

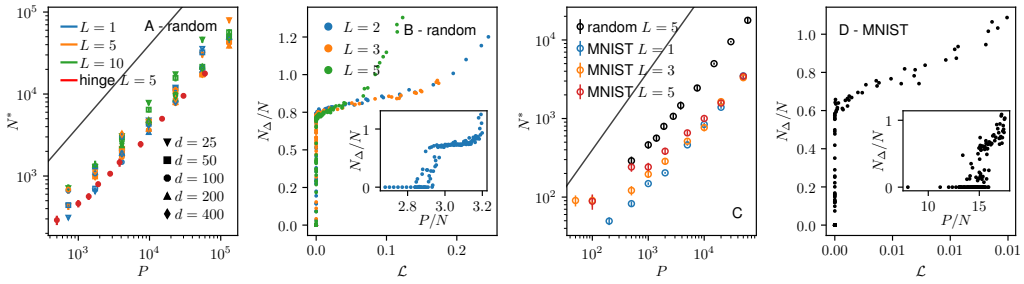


Figure 1: (A, B) Random data and (C, D) MNIST dataset. (A) and (C) depict the location N^* of the transition as a function of the number P, for networks with different cost functions or sizes. (B) and (D) show that in the $\mathcal{L}$-N_Δ/N and P/N-N_Δ/N planes the transition displays a discontinuous jump.

4 Generalization at and beyond jamming

In Fig. 2B the solid curve shows the error on the test set after training on the MNIST dataset (that is, after a fixed number of steps, see Appendix D for the details) and the dashed curve represents the value of the smallest error obtained during training, at prior time-steps (see Fig. 2A). The former displays a cusp at the transition point, indicating over-fitting [7]. The presence of the cusp is reminiscent of the teacher-student problems, where a cusp appears either because of noise in the teacher [36, 12, 1] or because of a mean-square error loss [28]. Strong over-fitting takes place only in the vicinity of the critical jamming transition (Fig. 2B-C), and beyond this point the accuracy keeps improving as the number of parameters increases [32, 31, 4], although it does so quite slowly. Curves can be fitted as $\epsilon_0 + \epsilon_1 \left(N/N^*\right)^{-\alpha}$ with $\alpha \approx 0.22$ (final test error) and $\alpha \approx 0.23$ (best test error). Understanding why generalization keeps slowly improving is a challenge for the future. Approaches studying the limit $h \to \infty$ may provide an interesting path forward [23, 30, 34]. Such a decay is at odds with the perceptron, where the accuracy asymptotically decreases with N. Furthermore, we posit that at fixed P the benefit of early stopping [33] should diminish in the large-size limit.

We have verified that the overall trends showed in Fig. 2 qualitatively hold also for other depths, see Appendix D.

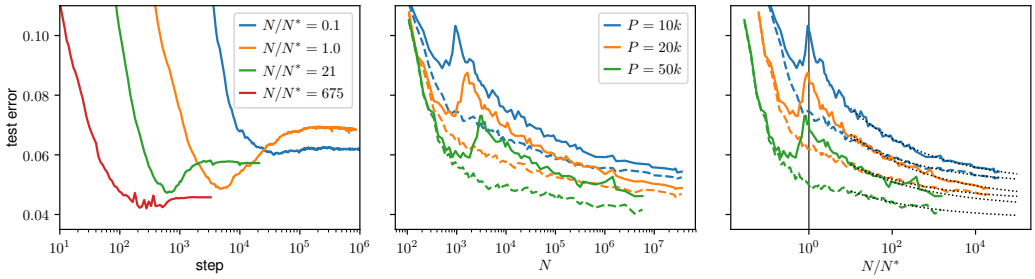


Figure 2: We trained a 5 hidden layer fully-connected network on MNIST. (A) Typical evolution of the generalization error over training time, for systems located at different points relatively to the jamming transition (for $P = 50k$): over-fitting is marked by the gap between the value at the end of training and the minimum at prior times. Notice that training of over-parametrized systems halts sooner because the networks have achieved zero loss over the training set. (B) Test error at the final point of training (solid line) and minimum error achieved during training (dashed line) vs. system size. (C) When N is scaled by $N^*(P)$ it is clear that over-fitting occurs at the jamming transition. We fit the error curves with functions of the form $\text{error} = \epsilon_0 + C(N/N^*)^{-\alpha}$, depicted in black dashed lines, $\alpha = 0.22$ for the last test error curves (plain lines) and $\alpha = 0.23$ for the best test error curves (dashed lines). Such curves can be fit equally well by a logarithmic behavior.

5 Conclusions

The hinge loss let us recast the minimization of a loss function as a constraint-satisfaction problem with continuous degrees of freedom. A similar approach was used in the field of interacting particles, which display a sharp jamming transition affecting the landscape if the interaction is chosen to be finite range [22]. Theoretical tools developed in that field allowed us to predict a sharp transition as the number of network parameters is varied, separating a region in the (P, N) plane where a global minimum can be found ($\mathcal{L} = 0$) from a region where the number of unsatisfied constraints is a fraction of the number of parameters, so $\mathcal{L} > 0$. These results also shed light on several aspects of deep learning:

Not getting stuck in local minima: In the over-parametrized regime, the dynamics does not get stuck in local minima because the number of constraints to satisfy is too small to hamper minimization. It follows from our assumptions on the negative eigenspace of the matrix $\mathcal{H}_p$ that in this regime the landscape is flat and local minima do not exist. This is the case for $P/N < r_c$, where r_c is $O(1)$.

Reference point for fitting and generalization: There exists a critical curve $N^*(P)$ on the N-P plane above which the global minima of the landscape become accessible. The curve also appears to be linked to the generalization potential of the model. We show that in the cases that we considered, *(i)* the generalization error decreases when $N \ll N^*$; then *(ii)* it increases and culminates in a cusp at $N \approx N^*$ that is erased by early stopping, most useful in this region; finally, *(iii)* in the over-parametrized phase, it monotonically decreases, although very slowly, an observation which remains unexplained from a theoretical point-of-view.

Acknowledgments

We thank Marco Baity-Jesi, Carolina Brito, Chiara Cammarota, Taco S. Cohen, Silvio Franz, Yann LeCun, Florent Krzakala, Riccardo Ravasio, Pierfrancesco Urbani and Lenka Zdeborova for helpful discussions. This work was partially supported by the grant from the Simons Foundation (#454935 Giulio Biroli, #454953 Matthieu Wyart). M.W. thanks the Swiss National Science Foundation for support under Grant No. 200021-165509. The manuscript [35], which appeared at the same time than ours, shows that the critical properties of the jamming transition found for the non-convex perceptron [14] hold more generally in some shallow networks. This universality is an intriguing result. Understanding the connection with our findings (see also [18]) is certainly worth future studies.

References

[1] Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

[2] Marco Baity-Jesi, Levent Sagun, Mario Geiger, Stefano Spigler, Gerard Ben Arous, Chiara Cammarota, Yann LeCun, Matthieu Wyart, and Giulio Biroli. Comparing dynamics: Deep neural networks versus glassy systems. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 314–323, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[3] Andrew J Ballard, Ritankar Das, Stefano Martiniani, Dhagash Mehta, Levent Sagun, Jacob D Stevenson, and David J Wales. Energy landscapes for machine learning. *Physical Chemistry Chemical Physics*, 2017.

[4] Yamini Bansal, Madhu Advani, David D Cox, and Andrew M Saxe. Minnorm training: an algorithm for training over-parameterized deep neural networks. *CoRR*, 2018.

[5] Ludovic Berthier and Giulio Biroli. Theoretical perspective on the glass transition and amorphous materials. *Reviews of Modern Physics*, 83(2):587, 2011.

[6] Siegfried Bös and Manfred Opper. Dynamics of training. In *Advances in Neural Information Processing Systems*, pages 141–147, 1997.

[7] Rich Caruana, Steve Lawrence, and C Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408, 2001.

[8] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

[9] Yaim Cooper. The loss landscape of overparameterized neural networks. *arXiv preprint arXiv:1804.10200*, 2018.

[10] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems*, pages 2933–2941, 2014.

[11] Gustavo Düring, Edan Lerner, and Matthieu Wyart. Phonon gap and localization lengths in floppy materials. *Soft Matter*, 9(1):146–154, 2013.

[12] Andreas Engel and Christian Van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, 2001.

[13] Silvio Franz, Sungmin Hwang, and Pierfrancesco Urbani. Jamming in multilayer supervised learning models. *arXiv preprint arXiv:1809.09945*, 2018.

[14] Silvio Franz and Giorgio Parisi. The simplest model of jamming. *Journal of Physics A: Mathematical and Theoretical*, 49(14):145001, 2016.

[15] Silvio Franz, Giorgio Parisi, Maxime Sevelev, Pierfrancesco Urbani, and Francesco Zamponi. Universality of the sat-unsat (jamming) threshold in non-convex continuous constraint satisfaction problems. *SciPost Physics*, 2(3):019, 2017.

[16] C Daniel Freeman and Joan Bruna. Topology and geometry of deep rectified network optimization landscapes. *International Conference on Learning Representations*, 2017.

[17] Xavier Gastaldi. Shake-shake regularization of 3-branch residual networks. *International Conference on Learning Representations*, 2017.

[18] Mario Geiger, Stefano Spigler, Stéphane d'Ascoli, Levent Sagun, Marco Baity-Jesi, Giulio Biroli, and Matthieu Wyart. The jamming transition as a paradigm to understand the loss landscape of deep neural networks. *arXiv preprint arXiv:1809.09349*, 2018.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, pages 1729–1739, 2017.

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456, 2015.

[22] Andrea J Liu, Sidney R Nagel, W Saarloos, and Matthieu Wyart. *The jamming scenario - an introduction and outlook*. OUP Oxford, 06 2010.

[23] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[25] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992.

[26] Florent Krzakala and Jorge Kurchan. Landscape analysis of constraint satisfaction problems. *Physical Review E*, 76(2):021122, 2007.

[27] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[28] Zhenyu Liao and Romain Couillet. The dynamics of learning: A random matrix approach. *arXiv preprint arXiv:1805.11917*, 2018.

[29] Zachary C Lipton. Stuck in a what? adventures in weight space. *International Conference on Learning Representations*, 2016.

[30] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layers neural networks. *arXiv preprint arXiv:1804.06561*, 2018.

[31] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

[32] Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*, 2017.

[33] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[34] Grant M Rotskoff and Eric Vanden-Eijnden. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *arXiv preprint arXiv:1805.00915*, 2018.

[35] P. Urbani S. Franz, S. Hwang. Jamming in multilayer supervised learning models. *arXiv preprint arXiv:1809.09945*, 2018.

[36] David Saad and Sara A Solla. On-line learning in soft committee machines. *Physical Review E*, 52(4):4225, 1995.

[37] Levent Sagun, Léon Bottou, and Yann LeCun. Singularity of the hessian in deep learning. *International Conference on Learning Representations*, 2017.

[38] Levent Sagun, Utku Evci, V. Uğur Güney, Yann Dauphin, and Léon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *ICLR 2018 Workshop Contribution, arXiv:1706.04454*, 2017.

[39] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *International Conference on Learning Representations*, 2014.

[40] Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

[41] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[42] Alexei V. Tkachenko and Thomas A. Witten. Stress propagation through frictionless granular material. *Phys. Rev. E*, 60(1):687–696, Jul 1999.

[43] Luca Venturi, Afonso Bandeira, and Joan Bruna. Neural networks with finite intrinsic dimension have no spurious valleys. *arXiv preprint arXiv:1802.06384*, 2018.

[44] Matthieu Wyart, Leonardo E Silbert, Sidney R Nagel, and Thomas A Witten. Effects of compression on the vibrational modes of marginally jammed solids. *Physical Review E*, 72(5):051306, 2005.

[45] Lenka Zdeborová and Florent Krzakala. Phase transitions in the coloring of random graphs. *Physical Review E*, 76(3):031131, 2007.

A Effective number of degrees of freedom

Due to several effects discussed in the main text, the function $f(\mathbf{x};\mathbf{W})$ can effectively depend on less variables that the number of parameters, and thus reduce the dimension of the space spanned by the gradients $\nabla_{\mathbf{W}} f(\mathbf{x};\mathbf{W})$ that enters in the theory. For instance, there could be symmetries that reduce the number of effective degrees of freedom (e.g. each ReLU activation function has one of such symmetries, since one can rescale inputs and outputs in such a way that the post-activation is left invariant); another reason could be that a neuron might never activate for all the training data, thus effectively reducing the number of neurons in the network; furthermore, we expect that the network's true dimension would also be reduced if its architecture presents some bottlenecks, is poorly designed or poorly initialized. For example if all biases are too negative on the neurons of one layer in the Relu case, the network does not transmit any signals, leading to $N = 1$ and to the possible absence of unstable directions even if the number of parameters is very large.

It is tempting to define the effective dimension by considering the dimension of the space spanned by $\nabla_{\mathbf{W}} f(\mathbf{x}_\mu;\mathbf{W})$ as μ varies. This definition is not practical for small number of samples P however, because this dimension would be bounded by P. We can overcome such a problem by considering a neighborhood of each point $\mathbf{x}_\mu$, where the network's function and its gradient can be expanded in the pattern space:

$$f(\mathbf{x}) \approx f(\mathbf{x}_\mu) + (\mathbf{x}-\mathbf{x}_\mu)\cdot\nabla_{\mathbf{x}} f(\mathbf{x}_\mu), \tag{4}$$

$$\nabla_{\mathbf{W}} f(\mathbf{x}) \approx \nabla_{\mathbf{W}} f(\mathbf{x}_\mu) + (\mathbf{x}-\mathbf{x}_\mu)\cdot\nabla_{\mathbf{x}}\nabla_{\mathbf{W}} f(\mathbf{x}_\mu). \tag{5}$$

Varying the pattern μ and the point $\mathbf{x}$ in the neighborhood of $\mathbf{x}_\mu$, we can build a family M of vectors:

$$M = \left\{\nabla_{\mathbf{W}} f(\mathbf{x}_\mu) + (\mathbf{x}-\mathbf{x}_\mu)\cdot\nabla_{\mathbf{x}}\nabla_{\mathbf{W}} f(\mathbf{x}_\mu)\right\}_{\mu,\mathbf{x}}. \tag{6}$$

We then define the effective dimension N_{eff} as the dimension of M. Because of the linear structure of M, it is sufficient to consider, for each μ, only $d+1$ values for x, e.g. $x - x_\mu = 0, \hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_d$, where $\hat{\mathbf{e}}_n$ is the unit vector along the direction n. The effective dimension is therefore

$$N_{\text{eff}} = \text{rk}(G), \tag{7}$$

where the elements of the matrix G are defined as

$$G_{i,\alpha} \equiv \partial_{W_i} f(\mathbf{x}_\mu) + \hat{\mathbf{e}}_n \cdot \nabla_{\hat{\mathbf{e}}_n}\partial_{W_i} f(\mathbf{x}_\mu), \tag{8}$$

with $\alpha \equiv (\mu, n)$. The index n ranges from 0 to d, and $\hat{\mathbf{e}}_0 \equiv 0$.

In Fig. 3 we show the effective number of parameters N_{eff} versus the total number of parameters N, in the case of a network with $L = 3$ layers trained on the first 10 PCA components of the MNIST dataset. There is no noticeable difference between the two quantities: the only reduction is due to the symmetries induced by the ReLU functions (there is one such symmetry per neuron. Indeed the ReLU function $\rho(z) = z\Theta(z)$ satisfies $\Lambda\rho(z/\Lambda) \equiv \rho(z)$.) We observed the same results for random data.

B $\text{sp}(H_p)$ is symmetric for ReLu activation functions and random data

We consider $\mathcal{H}_p = \sum_\mu y_\mu \rho\left(\Delta_\mu\right)\hat{\mathcal{H}}_\mu$, where $\hat{\mathcal{H}}_\mu$ is the Hessian of the network function $f(\mathbf{x}_\mu;\mathbf{W})$ and ρ is the Relu function. We want to argue that the spectrum of $\mathcal{H}_p$ is symmetric in the limit of large N.

First, we argue that it must be so for $\hat{\mathcal{H}}_\mu$. It is equivalent to show that $\text{tr}(\hat{\mathcal{H}}_\mu^n) = 0$ for any *odd* n.

$$\text{tr}(\hat{\mathcal{H}}_\mu^n) = \sum_{i_1,i_2,\ldots,i_n} \hat{\mathcal{H}}^\mu_{i_1,i_2}\hat{\mathcal{H}}^\mu_{i_2,i_3}\cdots\hat{\mathcal{H}}^\mu_{i_n,i_1}, \tag{9}$$

where the indices $i_1,\ldots,i_n$ stand for synapses connecting a pair of neurons (i.e. each index is associated with a synaptic weight $W^{(j)}_{\alpha,\beta}$: we are not writing all the explicit indexes for the sake of clarity). The term of the hessian obtained when differentiating with respect to weights $W^{(j)}_{\alpha,\beta}$ and $W^{(k)}_{\gamma,\delta}$ reads

$$\hat{\mathcal{H}}^{\mu;(jk)}_{\alpha\beta;\gamma\delta} = \sum_{\pi_0,\ldots,\pi_L} \theta(a^\mu_{L,\pi_L})\cdots\theta(a^\mu_{1,\pi_1})x^\mu_{\pi_0}\cdot\partial_{W^{(j)}_{\alpha,\beta}}\partial_{W^{(k)}_{\gamma,\delta}}\left[W^{(L+1)}_{\pi_L}W^{(L)}_{\pi_L,\pi_{L-1}}\cdots W^{(1)}_{\pi_1\pi_0}\right]. \tag{10}$$

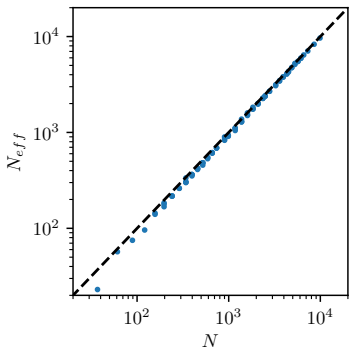


Figure 3: Results with the MNIST dataset, keeping the first 10 PCA components. $d = 10$ and $L = 3$, varying P and h. Effective N_{eff} vs total number of parameters N. N_{eff} is always smaller than N because there is a symmetry per each ReLU-neuron in the network.

Our argument is based on a symmetry of the problem (with random data): changing the sign of the weight of the last layer $W^{(L+1)} \longrightarrow -W^{(L+1)}$ and changing the labels $y_\mu \longrightarrow -y_\mu$ leaves the loss unchanged. We will show that this symmetry implies that $\mathrm{tr}(\hat{\mathcal{H}}_\mu^n)$ averaged over the random labels is zero for odd n. We find that the rank of $\hat{\mathcal{H}}_\mu$ is proportional to the number of active nodes. Therefore the spectrum of $\hat{\mathcal{H}}_\mu$ only contains $O(\sqrt{N})$ non zero eigenvalues. Since this number diverges we expect the spectrum to be self-averaging, even-though with two different scalings for the delta peak and the set of non-zero eigenvalues. In consequence, for any realisation of the data, odd moments are expected to be zero and the spectrum must be symmetric.

We thus have to show that $\mathrm{tr}(\hat{\mathcal{H}}_\mu^n)$ changes sign under the symmetry mentioned above for odd n. Note that the sum in Eq.10 contains a weight per each layer in the network, with the exception of the two layers j, k with respect to which we are deriving. This implies that any element of the hessian matrix where we have not differentiated with respect to the last layer ($j, k < L + 1$) is an odd function of the last layer $W^{(L+1)}$, meaning that if $W^{(L+1)} \longrightarrow -W^{(L+1)}$, then the sign of all these Hessian elements is inverted as well.

If in the argument of the sum in Eq. (9) there is no index belonging to the last layer, then the whole term changes sign under the transformation $W^{(L+1)} \longrightarrow -W^{(L+1)}$. Suppose now that, on the contrary, there are m terms with one index belonging to the last layer (we need not consider the case of two indices both belonging to the last layer because the corresponding term in the Hessian would be 0, as one can see in Eq. (10)). For each index equal to $L + 1$ (the last layer), there are exactly two terms: $\hat{\mathcal{H}}^\mu_{j,L+1}\hat{\mathcal{H}}^\mu_{L+1,k}$ (for some indexes j, k). Since j, k cannot be $L + 1$ too, this implies that the number m of terms with an index belonging to the last layer is always even. Consequently, when the sign of $W^{(L+1)}$ is reversed, the argument of the sum in Eq. (9) is multiplied by $(-1)^{n-m}$ (once for each term *without* an index belonging to the last layer), which is equal to -1 if n is odd, concluding our argument showing that $\hat{\mathcal{H}}_\mu$ has a symmetric spectrum. The same symmetry can be used to show that a matrix made of an odd product of matrices $\hat{\mathcal{H}}_\mu$, such as $\hat{\mathcal{H}}_\mu\hat{\mathcal{H}}_{\mu'}\hat{\mathcal{H}}_{\mu''}$, must also have a symmetric spectrum. These are the terms that contribute to $\mathrm{tr}(\mathcal{H}_p^n)$, which therefore it is also expected to vanish in the large N limit for all odd n.
Note that the sets of arguments presented above are not at a level of a rigorous proof, for which a careful analysis of sub-leading corrections would be needed.

C Density of pre-activations for ReLU activation functions

The densities of pre-activation (i.e. the value of the neurons before applying the activation function) is shown in Fig. 4) for random data. It contains a delta distribution in zero. The number N_c of pre-activations equal to zero when feeding a network $L = 5$ all its random dataset is $N_c \approx 0.21N$, corresponding to the number of directions in phase space where cusps are present in the loss function.

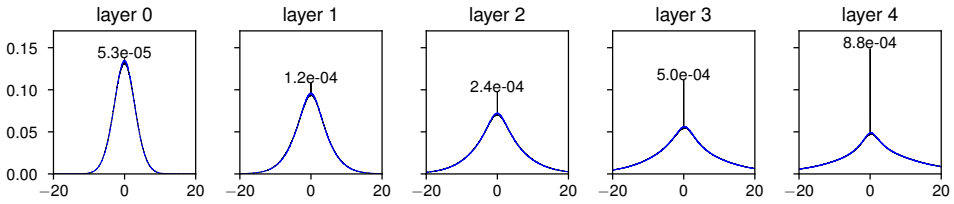


Figure 4: Density of the pre-activations for each layers with $L = 5$ and random data, averaged over all the runs just above the jamming transition with that architecture. Black: distribution obtained over the training set. Blue: previously unseen random data (the two curves are on top of each other except for the delta in zero). The values indicate the mass of the peak in zero, which is only present when the training set is considered.

For MNIST data we find $N_c \approx 0.19N$. By taking $L = 2$ and random data we find $N_c \approx 0.25N$. In these directions, stability can be achieved even if the hessian would indicate an instability. For this reason, instead of N_- in Equation 3 one should use $N/2 - N_c \approx 0.25N$.

D Parameters used in simulations

D.1 Random data

The dataset is composed of P points taken to lie on the d-dimensional hyper-sphere of radius $\sqrt{d}$, $\mathbf{x}_\mu \in \mathcal{S}^d$, with random label $y_\mu = \pm 1$. The networks are fully connected, and have an input layer of size d and L layers with h neurons each, culminating in a final layer of size 1. To find the transition we proceed as follows: we build a network with a number of parameters N large enough for it to be able to fit the whole dataset without errors. Next, we decrease the width h while keeping the depth L fixed, until the network cannot correctly classify all the data anymore within the chosen learning time. We denote this transition point N^*. As initial conditions for the dynamics we use the default initialization of `pytorch`: weights and biases are initialized with a uniform distribution on $[-\sigma, \sigma]$, where $\sigma^2 = 1/f_{in}$ and f_{in} is the number of incoming connections.

When using the cross entropy, the system evolves according to a stochastic gradient descent (SGD) with a learning rate of 10^{-2} for $5 \cdot 10^5$ steps and 10^{-3} for $5 \cdot 10^5$ steps (10^6 steps in total); the batch size is set to $\min(P/2, 1024)$, and batch normalization is used. We do not use any explicit regularization in training the networks. In Fig. 5 we check that $t = 10^6$ is enough to converge.

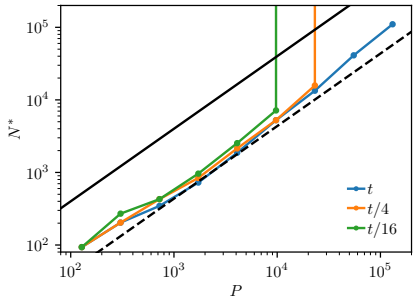


Figure 5: Convergence of the critical line for networks trained with cross entropy on random data.

When using the hinge loss, we use an orthogonal initialization [39], no batch normalization and $t = 2 \cdot 10^6$ steps of ADAM [24] with batch size P and a learning rate starting at 10^{-4}. In the experiments of section 3 (not for the experiments of section 4), we progressively divided the learning rate by 10 every 250k steps. Also in this case we do not use any explicit regularization in training the networks.

To observe the discontinuous jump in the number N_Δ of unsatisfied constraints at the transition (Fig. 1B and inset), we consider three architectures, both with $N \approx 8000$ and $d = h$ but with different

depths $L = 2$, $L = 3$ and $L = 5$. The vicinity of the transition is studied by varying P around the transition value and minimizing for 10^7 steps (a better minimization is needed to improve the precision close to the transition).

Details about Fig 1A hinge We took $d = h$ and trained for 2M steps. For some values of $P \in (500, 60\text{k})$, start at large h where we reach $N_\Delta = 0$ and decrease h until $N_\Delta > 0.1N$.

Details about Fig 1B We trained networks of depth 2,3,5 with $d = h = 62$, 51, 40 respectively for 10M steps. For $L = 3$ ($d = 51$, $h = 51$) we ran 128 training varying P from 21991 to 25918. For the value of N we take 7854 that correspond to the number of parameters minus the number of neurons, per neuron there is a degree of freedom lost in a symmetry induced by the homogeneity of the ReLU function. 37 of the runs have $N_\Delta = 0$, 74 have $N_\Delta > 0.4N$. Among the 19 remaining ones, 14 of them have N_Δ between 1 and 4, we think that these runs encounter numerical precision issues, we observed that using 32 bit precision accentuate this issue. We think that the 5 left with $4 < N_\Delta < 0.4N$ has been stoped too early. The same observation apply for the other depths.

D.2 Real data

The images in the MNIST dataset are gathered into two groups, with even and odd numbers and with labels $y_\mu = \pm 1$. The architecture of the network is as in the previous sections: the d inputs are fed to a cascade of L fully-connected layers with h neurons each, that in the end result in a single scalar output. The loss function used is always the hinge loss.

If we kept the original input size of $28 \times 28 = 784$ (each picture is 28×28 pixels) then the majority of the network's weights would be necessarily concentrated in the first layer (the width h cannot be too large in order to be able to compute the Hessian). To avoid this issue, we opt for a reduction of the input size. We perform a principal component analysis (PCA) on the whole dataset and we identify the 10 dimensions that carry the most variance on the whole dataset; then we use the components of each image along these directions as a new input of dimension $d = 10$. This projection hardly diminishes the performance of the network (which we find to be larger than 90% when using all the data and large N).

Details about Fig 1C We trained networks of depth 1,3,5 for 2M steps. For some values of $P \in (100, 50\text{k})$, start at large h where we reach $N_\Delta = 0$ and decrease h until $N_\Delta > 0.1N$.

Details about Fig 1D We trained a network of $L = 5$, $d = 10$, $h = 30$ for 3M steps. With P varying from 31k to 68k (using trainset and testset of MNIST).

Details about Fig 2 We trained a network of $L = 5$ and $d = 10$ for 500k steps. where $P \in \{10\text{k}, 20\text{k}, 50\text{k}\}$ and h varies from 1 to 3k. Fig 6 shows a comparison between $L = 5$ and $L = 2$.

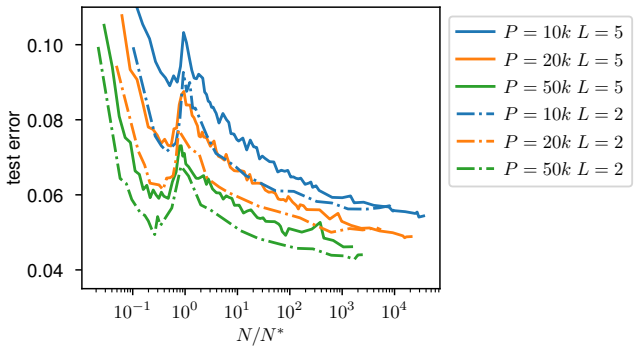


Figure 6: Generalization on MNIST 10 PCA. Comparison between two depth $L = 2$ and $L = 5$.