PAPER

Dynamical mean-field theory for stochastic gradient descent in Gaussian mixture classification*

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

PAPER: ML 2021

Dynamical mean-field theory for stochastic gradient descent in Gaussian mixture classification*

Francesca Mignacco[1,], Florent Krzakala[2,3], Pierfrancesco Urbani[1] and Lenka Zdeborová[1,4]**

[1] Institut de Physique Théorique, Université Paris-Saclay, CNRS, CEA, Gif-sur-Yvette, France
[2] Laboratoire de Physique, CNRS, École Normale Supérieure, PSL University, Paris, France
[3] IdePHICS Laboratory, EPFL, Switzerland
[4] SPOC Laboratory, EPFL, Switzerland
E-mail: francesca.mignacco@ipht.fr

Received 10 November 2021
Accepted for publication 14 November 2021
Published 29 December 2021

Online at stacks.iop.org/JSTAT/2021/124008
https://doi.org/10.1088/1742-5468/ac3a80

Abstract. We analyze in a closed form the learning dynamics of the stochastic gradient descent (SGD) for a single-layer neural network classifying a high-dimensional Gaussian mixture where each cluster is assigned one of two labels. This problem provides a prototype of a non-convex loss landscape with interpolating regimes and a large generalization gap. We define a particular stochastic process for which SGD can be extended to a continuous-time limit that we call stochastic gradient flow. In the full-batch limit, we recover the standard gradient flow. We apply dynamical mean-field theory from statistical physics to track the dynamics of the algorithm in the high-dimensional limit via a self-consistent

*This article is an updated version of: Mignacco F, Krzakala F, Urbani P and Zdeborová L 2020 Dynamical mean-field theory for stochastic gradient descent in Gaussian mixture classification *Advances in Neural Information Processing Systems* vol 33 ed H Larochelle, M Ranzato, R Hadsell, M F Balcan and H Lin (New York: Curran Associates) pp 9540–50.

**Author to whom any correspondence should be addressed.

1742-5468/21/124008+23$33.00

stochastic process. We explore the performance of the algorithm as a function of the control parameters shedding light on how it navigates the loss landscape.

Keywords: learning theory, machine learning

Contents

1. Introduction **2**

2. Setting and definitions **4**

3. Stochastic gradient-descent training dynamics **5**

4. Dynamical mean-field theory for SGF **6**

5. Results **10**

Acknowledgments 13

Appendix A. Derivation of the dynamical mean-field equations **13**

A.1. SUSY formulation 14

A.2. Saddle-point equations 17

A.3. Numerical solution of DMFT equations 19

Appendix B. Generalization error **19**

B.1. Perceptron with linear activation function 19

B.2. Perceptron with door activation function 20

Appendix C. Oracle error **20**

References **22**

1. Introduction

Understanding how stochastic gradient descent (SGD) manages to train artificial neural networks with good generalization capabilities by exploring the high-dimensional non-convex loss landscape is one of the central problems in the theory of machine learning. A popular attempt to explain this behavior is by showing that the loss landscape itself is simple, with no spurious (i.e. leading to bad test error) local minima. Some empirical evidence instead leads to the conclusion that the loss landscape of state-of-the-art deep neural networks actually has spurious local (or even global) minima and SGD is able to find them [1, 2]. Still, the SGD algorithm, initialized at random, leads to good generalization properties in practice. It became clear that a theory that would explain this success needs to account for the whole trajectory of the algorithm. Yet this remains a challenging task, certainly for the state-of-the art deep networks trained on real datasets.

J. Stat. Mech. (2021) 124008

Related work—A detailed description of the whole trajectory taken by the (stochastic) gradient descent has so far only been obtained in several special cases. The first such case is in deep linear networks where the dynamics of gradient descent have been analyzed [3, 4]. While this line of work has led to very interesting insights about the dynamics, linear networks lack the expressivity of the non-linear ones and the large-time behavior of the algorithm can be obtained with a simple spectral algorithm. Moreover, the analysis of dynamics in deep linear networks was not extended to the case of SGD. The second case where the trajectory of the algorithm was understood in detail is the *one-pass* (online) SGD for two-layer neural networks with a small hidden layer in the teacher–student setting [5–9]. However, the one-pass assumption made in those analyses is far from what is done in practice and is unable to access the subtle difference between the training and test error that leads to many of the empirical mysteries observed in deep learning. A third very interesting line of research that recently provided insight about the behavior of SGD concerns two-layer networks with divergingly wide hidden layers. This mean-field limit [10–12] maps the dynamics into the space of functions where its description is simpler and the dynamics can be written in terms of a closed set of differential equations. It is not clear yet whether this analysis can be extended in a sufficiently explicit way to deeper or finite width neural networks. The term *mean-field* has been used in several contexts in machine learning [13–18]. Note that the term in the aforementioned works refers to a variety of approximations and concepts. In this work, we use it with the same meaning as in [19–21]. Most importantly, the term mean-field in our case has nothing to do with the width of an eventual hidden layer. We refer to [22] for a broader methodological review of mean-field methods and their applications to neural networks.

Our present work, inscribed in the above line of research, offers the dynamical mean-field theory (DMFT) formalism [19–21] leading to a closed set of integro-differential equations to track the full trajectory of the gradient descent (stochastic or not) from random initial conditions in the high-dimensional limit for in-general non-convex losses. While in general the DMFT is a heuristic statistical physics method, it has been amenable to rigorous proof in some cases [23]. This is hence an important future direction for the case considered in the present paper. The DMFT has been recently applied to a high-dimensional inference problem in [24, 25], studying the spiked matrix-tensor model. However, this problem does not allow a natural way to study the SGD or to explore the difference between training and test errors. In particular, the spiked matrix-tensor model does not allow for the study of the so-called interpolating regime, where the loss function is optimized to zero while the test error remains positive. As such, its landscape is intrinsically different from supervised learning problems since in the former the spurious minima proliferate at high values of the loss while the good ones lie at the bottom of the landscape. Instead, deep networks have both spurious and good minima at 100% training accuracy and their landscape much closer resembles the one of continuous constraint satisfaction problems [26, 27].

Main contributions—We study a natural problem of supervised classification where the input data come from a high-dimensional Gaussian mixture of several clusters, and all samples in one cluster are assigned to one of two possible output labels. We then consider a single-layer neural network classifier with a general non-convex loss

J. Stat. Mech. (2021) 124008

function. We analyze a SGD algorithm in which, at each iteration, the batch used to compute the gradient of the loss is extracted at random, and we define a particular stochastic process for which SGD can be extended to a continuous-time limit that we call stochastic gradient flow (SGF). In the full-batch limit we recover the standard gradient flow (GF). We describe the high-dimensional limit of the randomly initialized SGF with the DMFT that leads to a description of the dynamics in terms of a self-consistent stochastic process that we compare with numerical simulations. In particular, we show that the finite batch size can have a beneficial effect on the test error and acts as an effective regularization that prevents overfitting.

2. Setting and definitions

In all what follows, we will consider the high-dimensional setting where the dimension of each point in the dataset is $d \to \infty$ and the size of the training set $n = \alpha d$, being α a control parameter that we keep of order one.

We consider a training set made of n points

$$\mathbf{X} = (\mathbf{x}_1, \dots \mathbf{x}_n)^\top \in \mathbb{R}^{n \times d} \quad \text{with labels} \quad \mathbf{y} = (y_1, \dots y_n)^\top \in \{+1, -1\}^n. \tag{1}$$

The patterns $\mathbf{x}_\mu$ are given by

$$\mathbf{x}_\mu = c_\mu \frac{\mathbf{v}^*}{\sqrt{d}} + \sqrt{\Delta}\, \mathbf{z}_\mu, \quad \mathbf{z}_\mu \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \ \mu = 1, \dots n. \tag{2}$$

Without loss of generality, we choose a basis where $\mathbf{v}^* = (1, 1, \dots 1) \in \mathbb{R}^d$.

Two-cluster dataset: We will illustrate our results on a two-cluster example where the coefficients c_μ are taken at random $c_\mu = \pm 1$ with equal probability. Therefore, one has two symmetric clouds of Gaussian points centered around two vectors $\mathbf{v}^*$ and $-\mathbf{v}^*$. The labels of the data points are fixed by $y_\mu = c_\mu$. If the noise level Δ of the number of samples is small enough, the two Gaussian clouds are linearly separable by a hyperplane, as specified in detail in [28], and therefore a single layer neural network is enough to perform the classification task in this case. We hence consider learning with the simplest neural network that classifies the data according to $\hat{y}_\mu(\mathbf{w}) = \mathrm{sgn}[\mathbf{w}^\top \mathbf{x}_\mu / \sqrt{d}]$.

Three-cluster dataset: We also consider an example of three clusters where a good generalization error cannot be obtained by separating the points linearly. In this case we define $c_\mu = 0$ with probability $1/2$, and $c_\mu = \pm 1$ with probability $1/2$. The labels are then assigned as

$$y_\mu = -1 \quad \text{if} \quad c_\mu = 0, \quad \text{and} \quad y_\mu = 1 \quad \text{if} \quad c_\mu = \pm 1. \tag{3}$$

Hence, one has three clouds of Gaussian points, two external and one centered in zero. In order to fit the data, we consider a single layer-neural network with the door activation function, defined as

$$\hat{y}_\mu(\mathbf{w}) = \mathrm{sgn}\left[\left(\frac{\mathbf{w}^\top \mathbf{x}_\mu}{\sqrt{d}} \right)^2 - L^2 \right]. \tag{4}$$

J. Stat. Mech. (2021) 124008

The onset parameter L could be learned, but we will instead fix it to a constant.

Loss function: We study the dynamics of learning by the empirical risk minimization of the loss

$$\mathcal{H}(\mathbf{w}) = \sum_{\mu=1}^{n} \ell\left[y_\mu \phi\left(\frac{\mathbf{w}^\top \mathbf{x}_\mu}{\sqrt{d}}\right)\right] + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \tag{5}$$

where we have added a ridge regularization term. The activation function ϕ is given by

$$\phi(x) = \begin{cases} x & \text{linear for the two} - \text{cluster dataset} \\ x^2 - L^2 & \text{door for the three} - \text{cluster dataset.} \end{cases} \tag{6}$$

The DMFT analysis is valid for a generic loss function ℓ. However, for concreteness, in the result section we will focus on the logistic loss $\ell(v) = \ln\left(1 + \mathrm{e}^{-v}\right)$. Note that in this setting the two-cluster dataset leads to convex optimization, with a unique minimum for finite λ, and implicit regularization for $\lambda = 0$ [29], and was analyzed in detail in [28, 30]. Still the performance of SGD with finite batch size cannot be obtained in *static* ways. The three-cluster dataset, instead, leads to a generically non-convex optimization problem, which can present many spurious minima with different generalization abilities when the control parameters such as Δ and α are changed. We note that our analysis can be extended to neural networks with a small hidden layer [31]. This would allow one to study the role of over-parametrization, but it is left for future work.

3. Stochastic gradient-descent training dynamics

Discrete SGD dynamics—We consider the discrete gradient-descent dynamics for which the weight update is given by

$$\mathrm{w}_j(t+\eta) = \mathrm{w}_j(t) - \eta\left[\lambda \mathrm{w}_j(t) + \sum_{\mu=1}^{n} s_\mu(t)\Lambda'\left(y_\mu, \frac{\mathbf{w}(t)^\top \mathbf{x}_\mu}{\sqrt{d}}\right)\frac{\mathrm{x}_{\mu,j}}{\sqrt{d}}\right], \tag{7}$$

where we have introduced the function $\Lambda(y,h) = \ell\left(y\phi\left(h\right)\right)$ and we have indicated with a prime the derivative with respect to h, i.e. $\Lambda'(y,h) = y\ell'\left(y\phi\left(h\right)\right)\phi'\left(h\right)$. We consider the following initialization of the weight vector $\mathbf{w}(0) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d R)$, where $R > 0$ is a parameter that tunes the average length of the weight vector at the beginning of the dynamics[5]. The variables $s_\mu(t)$ are i.i.d. binary random variables. Their discrete-time dynamics can be chosen in two ways:

- In classical **SGD**, when sampling *with replacement*, at iteration t one extracts the samples with the following probability distribution

$$s_\mu(t) = \begin{cases} 1 & \text{with probability} \quad b \\ 0 & \text{with probability} \quad 1-b \end{cases} \tag{8}$$

[5] The DMFT equations we derive can be easily generalized to the case in which the initial distribution over $\mathbf{w}$ is different. We only need it to be separable and independent of the dataset.

and $b \in (0,1]$. In this way for each time iteration one extracts on average $B = bn$ patterns at random on which the gradient is computed and therefore the batch size is given by B. Note that if $b = 1$ one recovers the full-batch gradient descent.

- **Persistent SGD** is defined by a stochastic process for $s_\mu(t)$ given by the following probability rules

$$\begin{aligned} \text{Prob}(s_\mu(t+\eta) = 1 | s_\mu(t) = 0) &= \frac{1}{\tau}\eta \\ \text{Prob}(s_\mu(t+\eta) = 0 | s_\mu(t) = 1) &= \frac{(1-b)}{b\tau}\eta, \end{aligned} \tag{9}$$

where $s_\mu(0)$ is drawn from the probability distribution (8). In this case, for each time slice one has on average $B = bn$ patterns that are *active* and enter in the computation of the gradient. The main difference with respect to the usual SGD is that one keeps the same patterns and the same minibatch for a characteristic time $\tau b/(1-b)$. Again, setting $b = 1$ one gets the full-batch gradient descent and all of the patterns are always active.

Stochastic gradient flow—To write the DMFT we consider the continuous-time dynamics defined by the $\eta \to 0$ limit. This limit is not well defined for the usual SGD dynamics described by the rule (8) and we instead consider its *persistent* version described by equation (9). In this case the stochastic process for $s_\mu(t)$ is well defined for $\eta \to 0$ and one can write a continuous-time equation as

$$\dot{\mathrm{w}}_j(t) = -\lambda \mathrm{w}_j(t) - \sum_{\mu=1}^{n} s_\mu(t) \Lambda' \left(y_\mu, \frac{\mathbf{w}(t)^\top \mathbf{x}_\mu}{\sqrt{d}} \right) \frac{\mathrm{x}_{\mu,j}}{\sqrt{d}}. \tag{10}$$

Again, for $b = 1$ one recovers the GF. We call equation (10) *stochastic gradient flow* (SGF).

4. Dynamical mean-field theory for SGF

We will now analyze the SGF in the infinite size limit $n \to \infty$, $d \to \infty$ with $\alpha = n/d$ and b and τ fixed and of order one. In order to do that, we use DMFT. The derivation of the DMFT equations is given in appendix A, but here we will just present the main steps. The derivation extends the one reported in [32] for the non-convex perceptron model [26] (motivated there as a model of glassy phases of hard spheres). The main differences of the present work with respect to [32] are that here we consider a finite-batch gradient descent and that our dataset is structured, while in [32] the derivation was done for full-batch gradient descent and random i.i.d. inputs and i.i.d. labels, i.e. a case where one cannot investigate generalization error and its properties. The starting point of the

J. Stat. Mech. (2021) 124008

DMFT is the dynamical partition function

$$Z_{\rm dyn} = \int_{\mathbf{w}(0)=\mathbf{w}^{(0)}} \mathcal{D}\mathbf{w}(t) \prod_{j=1}^{d} \delta \left[-\dot{\mathrm{w}}_j(t) - \lambda \mathrm{w}_j(t) - \sum_{\mu=1}^{n} s_\mu(t) \Lambda' \left(y_\mu, \frac{\mathbf{w}(t)^\top \mathbf{x}_\mu}{\sqrt{d}} \right) \frac{\mathrm{x}_{\mu,j}}{\sqrt{d}} \right], \tag{11}$$

where $\mathcal{D}\mathbf{w}(t)$ stands for the measure over the dynamical trajectories starting from $\mathbf{w}(0)$. Since $Z_{\rm dyn} = 1$ (it is just an integral of a Dirac delta function) [33], one can average directly $Z_{\rm dyn}$ over the training set, the initial condition and the stochastic processes of $s_\mu(t)$. We indicate this average with the brackets $\langle \cdot \rangle$. Hence, we can write

$$Z_{\rm dyn} = \left\langle \int \mathcal{D}\mathbf{w}(t) \mathcal{D}\hat{\mathbf{w}}(t) \mathrm{e}^{S_{\rm dyn}} \right\rangle, \tag{12}$$

where we have defined

$$S_{\rm dyn} = \sum_{j=1}^{d} \int_0^{+\infty} \mathrm{d}t \, i\hat{\mathrm{w}}_j(t) \left(-\dot{\mathrm{w}}_j(t) - \lambda \mathrm{w}_j(t) - \sum_{\mu=1}^{n} s_\mu(t) \Lambda' \left(y_\mu, \frac{\mathbf{w}(t)^\top \mathbf{x}_\mu}{\sqrt{d}} \right) \frac{\mathrm{x}_{\mu,j}}{\sqrt{d}} \right). \tag{13}$$

and we have introduced a set of fields $\hat{\mathbf{w}}(t)$ to produce the integral representation of the Dirac delta function. The average over the training set can be then performed explicitly, and the dynamical partition function $Z_{\rm dyn}$ is expressed as an integral of an exponential with extensive exponent in d:

$$Z_{\rm dyn} = \int \mathcal{D}\mathbf{Q} \, \mathcal{D}\mathbf{m} \, \mathrm{e}^{dS(\mathbf{Q},\mathbf{m})}, \tag{14}$$

where $\mathbf{Q}$ and $\mathbf{m}$ are two dynamical order parameters defined in the appendices. Therefore, the dynamics in the $d \to \infty$ limit satisfy a large deviation principle and we can approximate $Z_{\rm dyn}$ with its value at the saddle point of the action S. In particular, one can show that the saddle point equations for the parameters $\mathbf{Q}$ and $\mathbf{m}$ can be recast into a self-consistent stochastic process for a variable $h(t)$ related to the typical behavior of $\mathbf{w}(t)^\top \mathbf{z}_\mu / \sqrt{d}$, which evolves according to the stochastic equation:

$$\partial_t h(t) = -(\lambda + \hat{\lambda}(t)) h(t) - \sqrt{\Delta}\, s(t)\, \Lambda' \left(y(c), r(t) - Y(t) \right) + \int_0^t \mathrm{d}t' M_R(t,t') h(t') + \xi(t), \tag{15}$$

where we have denoted by $r(t) = \sqrt{\Delta} h(t) + m(t)(c + \sqrt{\Delta} h_0)$ and $m(t)$ is the *magnetization*, namely $m(t) = \mathbf{w}(t)^\top \mathbf{v}^* / d$. The details of the computation are provided in the appendices. There are several sources of stochasticity in equation (15). First, one has a dynamical noise $\xi(t)$ that is Gaussian distributed and characterized by the correlations

$$\langle \xi(t) \rangle = 0, \qquad \langle \xi(t) \xi(t') \rangle = M_C(t,t'). \tag{16}$$

J. Stat. Mech. (2021) 124008

Furthermore, the starting point $h(0)$ of the stochastic process is random and distributed according to

$$P(h(0)) = \mathrm{e}^{-h(0)^2/(2R)}/\sqrt{2\pi R}. \tag{17}$$

Moreover, one has to introduce a quenched Gaussian random variable h_0 with a mean zero and an average one. We recall that the random variable $c = \pm 1$ with equal probability in the two-cluster model, while $c = 0, \pm 1$ in the three-cluster one. The variable $y(c)$ is therefore $y(c) = c$ in the two-cluster case, and is given by equation (3) in the three-cluster one. Finally, one has a dynamical stochastic process $s(t)$ whose statistical properties are specified in equation (9). The magnetization $m(t)$ is obtained from the following deterministic differential equation

$$\partial_t m(t) = -\lambda m(t) - \mu(t), \quad m(0) = 0^+. \tag{18}$$

The stochastic process for $h(t)$, the evolution of $m(t)$, and the statistical properties of the dynamical noise $\xi(t)$ depend on a series of kernels that must be computed self-consistently and are given by

$$\begin{aligned}
\hat{\lambda}(t) &= \alpha\Delta \left\langle s(t)\Lambda''\left(y(c), r(t)\right)\right\rangle, \\
\mu(t) &= \alpha \left\langle s(t)\left(c + \sqrt{\Delta}h_0\right)\Lambda'\left(y(c), r(t)\right)\right\rangle, \\
M_C(t,t') &= \alpha\Delta \left\langle s(t)s(t')\Lambda'\left(y(c), r(t)\right)\Lambda'\left(y(c), r(t')\right)\right\rangle, \\
M_R(t,t') &= \alpha\Delta \frac{\delta}{\delta Y(t')}\langle s(t)\Lambda'(y(c), r(t))\rangle\Big|_{Y=0}.
\end{aligned} \tag{19}$$

In equation (19), the brackets denote the average over all the sources of stochasticity in the self-consistent stochastic process. Therefore, one needs to solve the stochastic process in a self-consistent way. Note that $Y(t)$ in equation (15) is set to zero and we need it only to define the kernel $M_R(t,t')$. The set of equations (15), (18) and (19) can be solved by a simple straightforward iterative algorithm. One starts with a guess for the kernels and then runs the stochastic process for $h(t)$ several times to update the kernels. The iteration is stopped when a desired precision on the kernels is reached [34].

Note that, in order to solve equations (15), (18) and (19), one needs to discretize time. In the results section 5, in order to compare our theoretical predictions with numerical simulations, we will take the time discretization of DMFT equal to the learning rate in the simulations. In the time-discretized DMFT, this allows us to extract the variables $s(t)$ either from (8) (SGD) or (9) (persistent SGD). In the former case, this provides an SGD-inspired discretization of the DMFT equations, which is also exact in the discrete time provided that the weight increments do not have higher-order terms than $\mathcal{O}(\eta)$.

Finally, once the self-consistent stochastic process is solved, one also has access to the dynamical correlation function $C(t,t') = \mathbf{w}(t)\cdot\mathbf{w}(t')/d$, encoded in the dynamical order parameter $\mathbf{Q}$ that appears in the large deviation principle of equation (14). The

J. Stat. Mech. (2021) 124008

correlation $C(t,t')$ concentrates on $d\to\infty$ and therefore is controlled by the equations

$$\begin{aligned}
\partial_t C(t',t) =& -\tilde{\lambda}(t)C(t,t') + \int_0^t \mathrm{d}s\, M_R(t,s)C(t',s) + \int_0^{t'} \mathrm{d}s\, M_C(t,s)R(t',s) \\
& - m(t')\left(\int_0^t \mathrm{d}s M_R(t,s)m(s) + \mu(t) - \hat{\lambda}(t)m(t)\right) \quad \text{if } t\neq t', \\
\frac{1}{2}\,\partial_t C(t,t) =& -\tilde{\lambda}(t)C(t,t) + \int_0^t \mathrm{d}s\, M_R(t,s)C(t,s) + \int_0^t \mathrm{d}s\, M_C(t,s)R(t,s) \\
& - m(t)\left(\int_0^t \mathrm{d}s\, M_R(t,s)m(s) + \mu(t) - \hat{\lambda}(t)m(t)\right), \\
\partial_t R(t,t') =& -\tilde{\lambda}(t)R(t,t') + \delta(t-t') + \int_{t'}^t \mathrm{d}s\, M_R(t,s)R(s,t'),
\end{aligned} \tag{20}$$

where we have used the shorthand notation $\tilde{\lambda}(t)=\lambda+\hat{\lambda}(t)$. We consider the linear response regime, and $R(t,t')=\sum_i \delta \mathrm{w}_i(t)/\delta H_i(t')/d$ is a response function that controls the variations of the weights when their dynamical evolution is affected by an infinitesimal local field $H_i(t)$. Coupling a local field $H_i(t)$ to each variable $w_i(t)$ changes the loss function as follows: $\mathcal{H}\left(\mathbf{w}(t)\right)\to\mathcal{H}\left(\mathbf{w}(t)\right)-\sum_{i=1}^d H_i(t)w_i(t)$, resulting in an extra term $H_i(t)$ in the right-hand side of equation (10). We then consider the limit $H_i(t)\to 0$. It is interesting to note that the second of equations. (20) controls the evolution of the norm of the weight vector $C(t,t)$ and even if we set $\lambda=0$ we get that it contains an effective regularization $\hat{\lambda}(t)$ that is dynamically self-generated [35].

Dynamics of the loss and the generalization error—Once the solution for the self-consistent stochastic process is found, one can get several interesting quantities. First, one can look at the training loss, which can be obtained as

$$e(t)=\alpha\langle\Lambda(y,r(t))\rangle, \tag{21}$$

where again the brackets denote the average over the realization of the stochastic process in equation (15). The training accuracy is given by

$$a(t)=1-\langle\theta(-y\phi(r(t)))\rangle \tag{22}$$

and, by definition, it is equal to one as soon as all vectors in the training set are correctly classified. Finally, one can compute the generalization error. At any time step, it is defined as the fraction of mislabeled instances:

$$\varepsilon_{\mathrm{gen}}(t)=\frac{1}{4}\mathbb{E}_{\mathbf{X},\mathbf{y},\mathbf{x}_{\mathrm{new}},y_{\mathrm{new}}}\left[\left(y_{\mathrm{new}}-\hat{y}_{\mathrm{new}}\left(\mathbf{w}(t)\right)\right)^2\right], \tag{23}$$

J. Stat. Mech. (2021) 124008

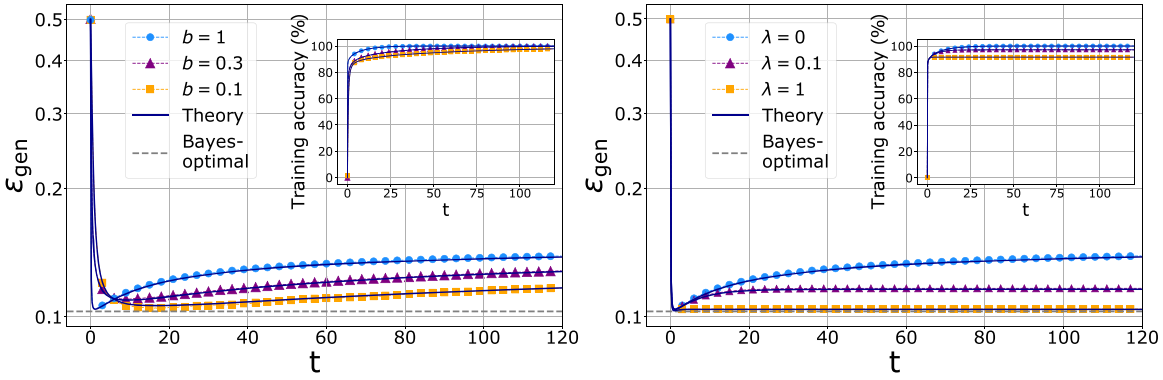

Figure 1. (Left) Generalization error as a function of the training time for persistent SGD in the two-cluster model, with $\alpha = 2$, $\Delta = 0.5$, $\lambda = 0$, $1/\tau = 0.6$ and different batch sizes $b = 1, 0.3, 0.1$. The continuous lines mark the numerical solution of DMFT equations, while the symbols are the results of simulations at $d = 500$, $\eta = 0.2$, and $R = 0.01$. The dashed gray line marks the Bayes-optimal error from [28]. (Right) Generalization error as a function of the training time for full-batch gradient descent in the two-cluster model with different regularization $\lambda = 0, 0.1, 1$ and the same parameters as in the left panel. In each panel, the inset shows the training accuracy as a function of the training time.

where $\{\mathbf{X}, \mathbf{y}\}$ is the training set, $\mathbf{x}_{\text{new}}$ is an unseen data point and $\hat{y}_{\text{new}}$ is the estimator for the new label y_{new}. The dependence on the training set here is hidden in the weight vector $\mathbf{w}(t) = \mathbf{w}(t, \mathbf{X}, \mathbf{y})$. In the two-cluster case, one can easily show that

$$\varepsilon_{\text{gen}}(t) = \frac{1}{2}\,\text{erfc}\left(\frac{m(t)}{\sqrt{2\Delta\, C(t,t)}}\right). \tag{24}$$

Conversely, for the door activation trained on the three-cluster dataset we obtain

$$\varepsilon_{\text{gen}}(t) = \frac{1}{2}\,\text{erfc}\left(\frac{L}{\sqrt{2\Delta C(t,t)}}\right) + \frac{1}{4}\left(\text{erf}\left(\frac{L - m(t)}{\sqrt{2\Delta C(t,t)}}\right) + \text{erf}\left(\frac{L + m(t)}{\sqrt{2\Delta C(t,t)}}\right)\right). \tag{25}$$

5. Results

In this section, we compare the theoretical curves resulting from the solution of the DMFT equations derived in section 4 to numerical simulations. This analysis allows us to gain insight into the learning dynamics of SGD and its dependence on the various control parameters in the two models under consideration.

The left panel of figure 1 shows the learning dynamics of the persistent-SGD algorithm in the two-cluster model without regularization $\lambda = 0$. We clearly see a good match between the numerical simulations and the theoretical curves obtained from DMFT, also notably for small values of batch size b and dimension $d = 500$. The figure shows that regions exist in control parameter space where persistent SGD is able to reach 100% training accuracy, while the generalization error is bounded away from zero.

J. Stat. Mech. (2021) 124008

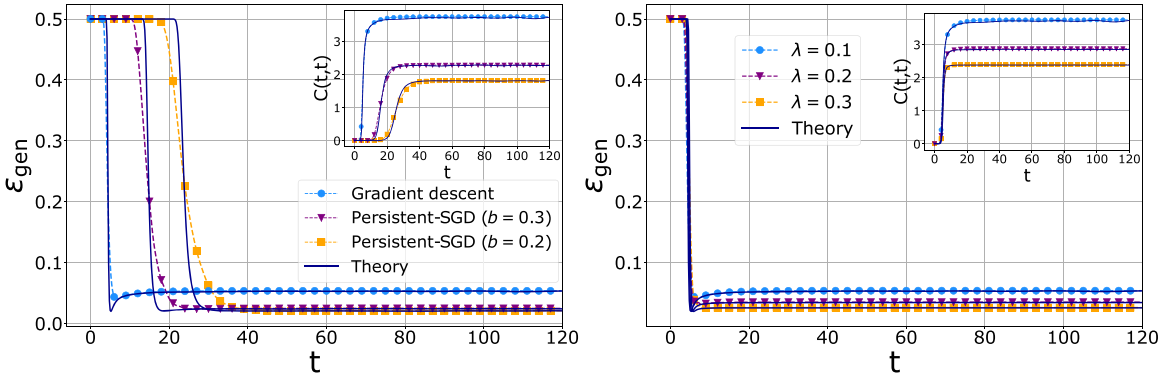

Figure 2. (Left) Generalization error as a function of the training time in the three-cluster model, at fixed $\alpha = 3$, $\Delta = 0.05$, $L = 0.7$, $\lambda = 0.1$, for full-batch gradient descent and persistent SGD with different batch sizes $b = 0.2, 0.3$ and activation rate $1/\tau = b$. The continuous lines mark the numerical solution of DMFT equations, while the symbols represent simulations at $\eta = 0.2$, $R = 0.01$, and $d = 5000$. (Right) Generalization error as a function of training time for full-batch gradient descent in the three-cluster model, at fixed $\alpha = 3$, $\Delta = 0.05$, $L = 0.7$, $\eta = 0.2$, $R = 0.01$, and different regularizations $\lambda = 0.1, 0.2, 0.3$. The simulations are done at $d = 5000$. In each panel, the inset shows the norm of the weights as a function of the training time.

Remarkably, we observe that the additional noise introduced by decreasing the batch size b results in a shift of the early-stopping minimum of the generalization error at larger times and that, in the time window we show, a batch size smaller than one has a beneficial effect on the generalization error at long times. The right panel illustrates the role of regularization in the same model trained with full-batch gradient descent, presenting that regularization has a similar influence on the learning curve as small batch-size but without the slow-down incurred by persistent SGD.

The influence of the batch size b and the regularization λ for the three-cluster model is shown in figure 2. We see an analogous effect as for the two-clusters in figure 1. In the inset of figure 2, we show the norm of the weights as a function of the training time. Both with the smaller mini-batch size and larger regularization the norm is small, testifying further that the two play a similar role in this case.

One difference between the two-cluster and the three-cluster models we observe concerns the behavior of the generalization error at small times. Actually, for the three-cluster model, good generalization is reached because of finite-size effects. Indeed, the corresponding loss function displays a $\mathbb{Z}_2$ symmetry according to which for each local minimum $\mathbf{w}$ there is another one $-\mathbf{w}$ with exactly the same properties. Note that this symmetry is inherited from the activation function ϕ (6), which is even. This implies that if $d \to \infty$, the generalization error would not move away from 0.5 in finite time. However, when d is large but finite, at time $t = 0$ the weight vector has a finite projection on $\mathbf{v}^*$, which is responsible for the dynamical symmetry breaking and eventually for a low generalization error at long times. In order to obtain an agreement between the theory and simulations, we initialize $m(t)$ in the DMFT equations with its corresponding finite-d average value at $t = 0$. In the left panel of figure 3, we show

J. Stat. Mech. (2021) 124008

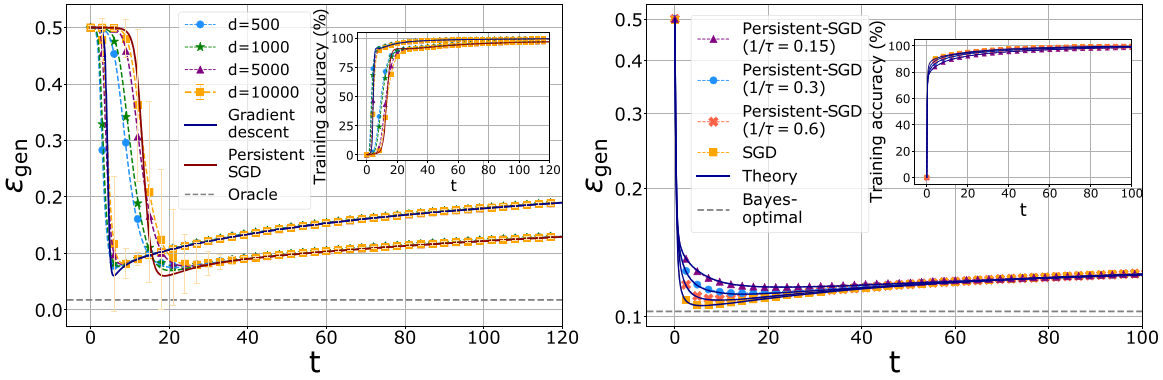

Figure 3. (Left) Generalization error as a function of the training time for full-batch gradient descent and persistent SGD with $1/\tau = b = 0.3$ in the three-cluster model, at fixed $\alpha = 2$, $\Delta = 0.05$, $L = 0.7$ and $\lambda = 0$. The continuous lines mark the numerical solution of DMFT equations, the symbols represent simulations at $\eta = 0.2$, $R = 1$, and increasing dimensions $d = 500, 1000, 5000, 10\,000$. Error bars are plotted for $d = 10\,000$. The dashed lines mark the oracle error (see appendices). (Right) Generalization error as a function of the training time for persistent SGD with different activation rates $1/\tau = 0.15, 0.3, 0.6$ and classical SGD in the two-cluster model, both with $b = 0.3$, $\alpha = 2$, $\Delta = 0.5$, $\lambda = 0$, $\eta = 0.2$, $R = 0.01$. The continuous lines mark the numerical solution of DMFT equations (in case of SGD we use the SGD-inspired discretization), while the symbols represent simulations at $d = 500$. The dashed lines mark the Bayes-optimal error from [28]. In each panel, the inset displays the training accuracy as a function of time.

that while this produces a small discrepancy at intermediate times that diminishes with growing size, at longer times the DMFT perfectly tracks the evolution of the algorithm.

The right panel of figure 3 summarizes the effect of the characteristic time τ in the persistent SGD, related to the typical persistence time of each pattern in the training mini-batch. When τ decreases, the persistent SGD algorithm is observed to be getting a better early-stopping generalization error and the dynamics get closer to the usual SGD dynamics. As expected, the $\tau \to \eta/b$ limit of the persistent SGD converges to the SGD. The SGD-inspired discretization of the DMTF equations shows a perfect agreement with the numerics.

Figure 4 presents the influence of the weight norm at initialization R on the dynamics, for the two-cluster (left) and three-cluster (right) model. For the two-cluster case, the gradient descent algorithm with all-zeros initialization 'jumps' on the Bayes-optimal error at the first iteration as derived in [28], and in this particular setting the generalization error monotonically increases in time. As R increases the early stopping error gets worse. At large times all of the initializations converge to the same value of the error, as they must, since this is a full-batch gradient descent without regularization that at large times converges to the max-margin estimator according to [29]. For the three-cluster model we observe a qualitatively similar behavior.

J. Stat. Mech. (2021) 124008

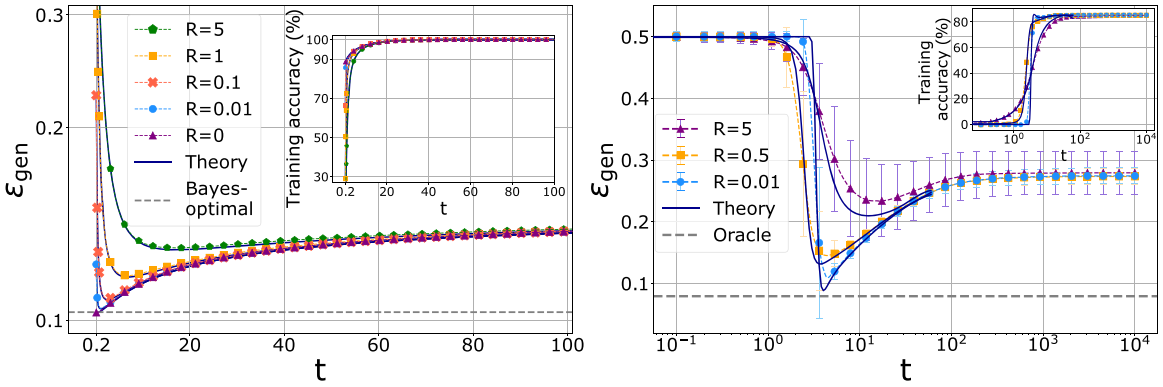

Figure 4. (Left) Generalization error as a function of training time for full-batch gradient descent in the two-cluster model, at fixed $\alpha = 2$, $\Delta = 0.5$, $\lambda = 0$, $\eta = 0.2$, and different initialization variances $R = 0, 0.01, 0.1, 1, 5$. The continuous lines mark the numerical solution of DMFT equations, while the symbols represent simulations at $d = 500$. The dashed lines mark the Bayes-optimal error from [28]. The $y-$axis is cut for better visibility. (Right) Generalization error as a function of training time for full-batch gradient descent in the three-cluster model, at fixed $\alpha = 3$, $\Delta = 0.1$, $\lambda = 0$, $\eta = 0.1$ and different initialization variances $R = 0.01, 0.5, 5$. The continuous lines mark the numerical solution of DMFT equations, while the symbols represent simulations at $d = 1000$. The dashed gray line marks the oracle error (see appendices). In each panel, the inset shows the training accuracy as a function of time.

Acknowledgments

This work was supported by 'Investissements d'Avenir' LabExPALM (ANR-10-LABX-0039-PALM), the ERC under the European Union's Horizon 2020 Research and Innovation Program 714608-SMiLe, as well as by the French Agence Nationale de la Recherche under Grant ANR-17-CE23-0023-01 PAIL and ANR-19-P3IA-0001 PRAIRIE.

Appendix A. Derivation of the dynamical mean-field equations

The derivation of the self-consistent stochastic process discussed in the main text can be obtained using tools of statistical physics of disordered systems. In particular, it has been done very recently for a related model, the spherical perceptron with random labels, in [32]. Our derivation extends the known DMFT equations by including

- Structure in the data;
- A stochastic version of gradient descent as discussed in the main text;
- The relaxation of the spherical constraint over the weights and the introduction of a ridge regularization term.

There are at least two ways to write the DMFT equations. One is by using field-theoretical techniques; otherwise one can employ a dynamical version of the so-called

J. Stat. Mech. (2021) 124008

cavity method [19]. Here, we opt for the first option that is generically very compact and immediate and it has a form that very much resembles a *static* treatment of the Gibbs measure of the problem [36]. We use a supersymmetric (SUSY) representation to derive the DMFT equations [32, 37]. We do not report all of the details, as they can be found in [32] along with an alternative derivation based on the cavity method, but we limit ourselves to providing the main points. We first consider the dynamical partition function, corresponding to equation (11) in the main text

$$Z_{\rm dyn} = \left\langle \int \left[\frac{\mathrm{d}\mathbf{w}^{(0)}}{(2\pi)^{\frac{d}{2}}}\,\mathrm{e}^{-\frac{1}{2}\|\mathbf{w}^{(0)}\|_2^2}\right]\int_{\mathbf{w}(0)=\mathbf{w}^{(0)}}\mathcal{D}\mathbf{w}(t) \right. \\ \left. \times\prod_{j=1}^{d}\delta\left[-\dot{\mathrm{w}}_j(t)-\lambda \mathrm{w}_j(t)-\sum_{\mu=1}^{n} s_\mu(t)\Lambda'\left(y_\mu,\frac{\mathbf{w}(t)^\top\mathbf{x}_\mu}{\sqrt{d}}\right)\frac{\mathrm{x}_{\mu,j}}{\sqrt{d}}\right]\right\rangle, \tag{A.1}$$

where the brackets $\langle\cdot\rangle$ stand for the average over $s_\mu(t)$, y_μ and the realization of the noise in the training set. The average over the initial condition is written explicitly. Note that we choose an initial condition that is Gaussian, but we could have chosen a different probability measure over the initial configuration of the weights. The equations can be generalized to other initial conditions as soon as they do not depend on quenched random variables that enter in the SGD dynamics and their distribution is separable. As observed in the main text, we have that $Z_{\rm dyn} = \langle Z_{\rm dyn}\rangle = 1$. We can write the integral representation of the Dirac delta function in equation (A.1) by introducing a set of fields $\hat{\mathbf{w}}(t)$

$$Z_{\rm dyn} = \left\langle \int \mathcal{D}\mathbf{w}(t)\mathcal{D}\hat{\mathbf{w}}(t)\mathrm{e}^{S_{\rm dyn}}\right\rangle, \tag{A.2}$$

where the dynamical action $S_{\rm dyn}$ is defined as in equation (13) of the main text

$$S_{\rm dyn} = \sum_{j=1}^{d}\int_0^{+\infty}\mathrm{d}t\, i\hat{\mathrm{w}}_j(t)\left(-\dot{\mathrm{w}}_j(t)-\lambda \mathrm{w}_j(t)-\sum_{\mu=1}^{n} s_\mu(t)\Lambda'\left(y_\mu,\frac{\mathbf{w}(t)^\top\mathbf{x}_\mu}{\sqrt{d}}\right)\frac{\mathrm{x}_{\mu,j}}{\sqrt{d}}\right). \tag{A.3}$$

A.1. SUSY formulation

The dynamical action $S_{\rm dyn}$ (A.3) can be rewritten in a SUSY form, by extending the time coordinate to include two Grassman coordinates θ and $\bar{\theta}$, i.e. $t_a \to a = (t_a, \theta_a, \bar{\theta}_a)$. The dynamic variable $\mathbf{w}(t_a)$ and the auxiliary variable $i\hat{\mathbf{w}}(t_a)$ are encoded in a super-field

$$\mathbf{w}(a) = \mathbf{w}(t_a) + i\theta_a\bar{\theta}_a\hat{\mathbf{w}}(t_a). \tag{A.4}$$

J. Stat. Mech. (2021) 124008

From the properties of Grassman variables [38]

$$\begin{aligned} \theta^2 = \bar{\theta}^2 = \theta\bar{\theta} + \bar{\theta}\theta = 0, \\ \int \mathrm{d}\theta = \int \mathrm{d}\bar{\theta} = 0, \qquad \int \mathrm{d}\theta\, \theta = \int \mathrm{d}\bar{\theta}\, \bar{\theta} = 1, \\ \partial_\theta g(\theta) = \int \mathrm{d}\theta\, g(\theta) \quad \text{for a generic function } g, \end{aligned} \tag{A.5}$$

it follows that

$$\int \mathrm{d}a f\left(\mathbf{w}(a)\right) = \int_0^{+\infty} \mathrm{d}t_a\, i\hat{\mathbf{w}}(t_a) f'\left(\mathbf{w}(t_a)\right). \tag{A.6}$$

We can use equation (A.6) to rewrite S_{dyn}. We obtain

$$S_{\mathrm{dyn}} = -\frac{1}{2}\int \mathrm{d}a\, \mathrm{d}b \mathcal{K}(a,b)\mathbf{w}(a)^\top \mathbf{w}(b) - \sum_{\mu=1}^{n} \int \mathrm{d}a\, s_\mu(a)\, \Lambda\left(y_\mu, h_\mu(a)\right), \tag{A.7}$$

where we have defined $h_\mu(a) \equiv \mathbf{w}(a)^\top \mathbf{x}_\mu/\sqrt{d}$ and we have implicitly defined the kernel $\mathcal{K}(a,b)$ such that

$$-\frac{1}{2}\int \mathrm{d}a\, \mathrm{d}b \mathcal{K}(a,b)\mathbf{w}(a)^\top \mathbf{w}(b) = \sum_{j=1}^{d} \int_0^{+\infty} \mathrm{d}t\, i\hat{\mathrm{w}}_j(t)\left(-\dot{\mathrm{w}}_j(t) - \lambda \mathrm{w}_j(t)\right). \tag{A.8}$$

By inserting the definition of $h_\mu(a)$ in the partition function, we have

$$\begin{aligned} Z_{\mathrm{dyn}} = \Bigg\langle \int \mathcal{D}\mathbf{w}(a)\mathcal{D}h_\mu(a)\mathcal{D}\hat{h}_\mu(a) \exp\Bigg[-\frac{1}{2}\int \mathrm{d}a\, \mathrm{d}b \mathcal{K}(a,b)\mathbf{w}(a)^\top \mathbf{w}(b) \\ - \sum_{\mu=1}^{n} \int \mathrm{d}a\, s_\mu(a)\, \Lambda\left(y_\mu, h_\mu(a)\right)\Bigg] \exp\left[\sum_{\mu=1}^{n} \int \mathrm{d}a\, i\, \hat{h}_\mu(a)\left(h_\mu(a) - \frac{\mathbf{w}(a)^\top \mathbf{x}_\mu}{\sqrt{d}}\right)\right]\Bigg\rangle. \end{aligned} \tag{A.9}$$

J. Stat. Mech. (2021) 124008

Let us consider the last factor in the integral in (A.9). We can perform the average over the random vectors $\mathbf{z}_\mu \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, denoted by an overline, as

$$\begin{aligned}
&\overline{\exp\left[\sum_{\mu=1}^{n} \int \mathrm{d}a\, i\hat{h}_\mu(a)\left(h_\mu(a) - \frac{\mathbf{w}(a)^\top \mathbf{x}_\mu}{\sqrt{d}}\right)\right]} \\
&\quad = \exp\left[\sum_{\mu=1}^{n} \int \mathrm{d}a\, i\hat{h}_\mu(a)\left(h_\mu(a) - c_\mu m(a) - \sqrt{\frac{\Delta}{d}}\mathbf{w}(a)^\top \mathbf{z}_\mu\right)\right] \\
&\quad = \exp\left[\sum_{\mu=1}^{n} \int \mathrm{d}a\, i\hat{h}_\mu(a)\left(h_\mu(a) - c_\mu m(a)\right) - \frac{\Delta}{2}\sum_{\mu=1}^{n} \int \mathrm{d}a\, \mathrm{d}b\, Q(a,b)\hat{h}_\mu(a)\hat{h}_\mu(b)\right],
\end{aligned} \tag{A.10}$$

where we have defined

$$\begin{aligned}
m(a) &= \frac{1}{d}\mathbf{w}(a)^\top \mathbf{v}^*, \\
Q(a,b) &= \frac{1}{d}\mathbf{w}(a)^\top \mathbf{w}(b).
\end{aligned} \tag{A.11}$$

By inserting the definitions of $m(a)$ and $Q(a,b)$ in the partition function, we obtain

$$Z_{\mathrm{dyn}} = \int \mathcal{D}\mathbf{Q}\, \mathcal{D}\mathbf{m}\, \mathrm{e}^{dS(\mathbf{Q},\mathbf{m})}, \tag{A.12}$$

where $\mathbf{Q} = \{Q(a,b)\}_{a,b}$, $\mathbf{m} = \{m(a)\}_a$ and

$$\begin{aligned}
S(\mathbf{Q},\mathbf{m}) &= \frac{1}{2}\log\det\left(Q(a,b) - m(a)m(b)\right) - \frac{1}{2}\int \mathrm{d}a\, \mathrm{d}b\, \mathcal{K}(a,b)Q(a,b) + \alpha \log \mathcal{Z}, \\
\mathcal{Z} &= \left\langle \int \mathcal{D}h(a)\mathcal{D}\hat{h}(a) \exp\left[-\frac{\Delta}{2}\int \mathrm{d}a\, \mathrm{d}b\; Q(a,b)\hat{h}(a)\hat{h}(b)\right.\right. \\
&\qquad \left.\left. + \int \mathrm{d}a\, i\hat{h}(a)\left(h(a) - cm(a)\right) - \int \mathrm{d}a\, s(a)\,\Lambda\left(y, h(a)\right)\right]\right\rangle.
\end{aligned} \tag{A.13}$$

We have used the fact that the samples are i.i.d. and removed the index $\mu = 1, \ldots n$. The brackets denote the average over the random variable c that has the same distribution as the c_μ, over y, distributed as y_μ, and over the random process of $s(t)$, defined by equation (9) in the main text. If we perform the change of variable $Q(a,b) \leftarrow Q(a,b) + m(a)m(b)$, we obtain

J. Stat. Mech. (2021) 124008

$$S(\mathbf{Q},\mathbf{m}) = \frac{1}{2}\log\det Q(a,b) - \frac{1}{2}\int \mathrm{d}a\,\mathrm{d}b\mathcal{K}(a,b)\left(Q(a,b)+m(a)m(b)\right) + \alpha\,\log\,\mathcal{Z},$$
$$\mathcal{Z} = \left\langle \int \mathcal{D}h(a)\mathcal{D}\hat{h}(a)\,\mathrm{e}^{S_{\mathrm{loc}}}\right\rangle, \tag{A.14}$$

where the effective local action S_{loc} is given by

$$\begin{aligned} S_{\mathrm{loc}} = &-\frac{\Delta}{2}\int \mathrm{d}a\,\mathrm{d}bQ(a,b)\hat{h}(a)\hat{h}(b) - \frac{\Delta}{2}\left(\int \mathrm{d}a\,\hat{h}(a)m(a)\right)^2 \\ &+ \int \mathrm{d}a\,i\hat{h}(a)\left(h(a)-cm(a)\right) - \int \mathrm{d}a\,s(a)\,\Lambda\left(y,h(a)\right). \end{aligned} \tag{A.15}$$

Performing a Hubbard–Stratonovich transformation on $\exp\left[-\frac{\Delta}{2}\left(\int \mathrm{d}a\,\hat{h}(a)m(a)\right)^2\right]$ and a set of transformations on the fields $h(a)$, we obtain that we can rewrite $\mathcal{Z}$ as

$$\begin{aligned} \mathcal{Z} = \Bigg\langle \int \frac{\mathrm{d}h_0}{\sqrt{2\pi}}\mathrm{e}^{-\frac{h_0^2}{2}} \int \mathcal{D}h(a)\mathcal{D}\hat{h}(a)\,\exp\Bigg[-\frac{1}{2}\int \mathrm{d}a\mathrm{d}b\;Q(a,b)\hat{h}(a)\hat{h}(b) \\ + \int \mathrm{d}a\,i\hat{h}(a)h(a) - \int \mathrm{d}a\,s(a)\,\Lambda\left(y,\sqrt{\Delta}h(a)+m(a)(c+\sqrt{\Delta}h_0)\right)\Bigg]\Bigg\rangle. \end{aligned} \tag{A.16}$$

A.2. Saddle-point equations

We are interested in the large d limit of Z_{dyn}, in which, according to equation (A.12), the partition function is dominated by the saddle-point value of $S(\mathbf{Q},\mathbf{m})$:

$$\begin{cases} \left.\dfrac{\delta S(\mathbf{Q},\mathbf{m})}{\delta Q(a,b)}\right|_{(\mathbf{Q},\mathbf{m})=(\tilde{\mathbf{Q}},\tilde{\mathbf{m}})} = 0 \\ \left.\dfrac{\delta S(\mathbf{Q},\mathbf{m})}{\delta m(a)}\right|_{(\mathbf{Q},\mathbf{m})=(\tilde{\mathbf{Q}},\tilde{\mathbf{m}})} = 0 \end{cases}. \tag{A.17}$$

$\tilde{Q}(a,b)$ is obtained from the equation

$$-\mathcal{K}(a,b) + Q^{-1}(a,b) + \frac{2\alpha}{\mathcal{Z}}\frac{\delta\mathcal{Z}}{\delta Q(a,b)} = 0. \tag{A.18}$$

The saddle-point equation for $\tilde{m}(a)$ is instead

$$-\int \mathrm{d}b\,\mathcal{K}(a,b)m(b) + \frac{\alpha}{\mathcal{Z}}\frac{\delta\mathcal{Z}}{\delta m(a)} = 0. \tag{A.19}$$

J. Stat. Mech. (2021) 124008

It can be easily shown by exploiting the Grassmann structure of equations (A.18) and (A.19) that they lead to a self-consistent stochastic process described by

$$\dot{h}(t) = -\tilde{\lambda}(t)h(t) - \sqrt{\Delta}s(t)\Lambda'\left(y, r(t) - Y(t)\right) + \int_0^t \mathrm{d}t' M_R(t,t')h(t') + \xi(t), \tag{A.20}$$

where the initial condition is drawn from $P(h(0)) \sim \mathrm{e}^{-h(0)^2/(2R)}/\sqrt{2\pi}$, and $r(t) = \sqrt{\Delta}h(t) + m(t)(c + \sqrt{\Delta}h_0)$, with $P_0(h_0) \sim \mathrm{e}^{-h_0^2/2}/\sqrt{2\pi}$. We have defined the auxiliary functions

$$\begin{aligned}
\mu(t) &= \alpha\left\langle s(t)\left(c + \sqrt{\Delta}h_0\right)\Lambda'\left(y, r(t)\right)\right\rangle, \\
\hat{\lambda}(t) &= \alpha\Delta\left\langle s(t)\Lambda''\left(y, r(t)\right)\right\rangle, \\
\tilde{\lambda}(t) &= \lambda + \hat{\lambda}(t),
\end{aligned} \tag{A.21}$$

and kernels

$$\begin{aligned}
M_C(t,t') &= \alpha\Delta\left\langle s(t)s(t')\Lambda'\left(y, r(t)\right)\Lambda'\left(y, r(t')\right)\right\rangle, \\
M_R(t,t') &= \alpha\Delta^{3/2}\left\langle s(t)s(t')\Lambda'\left(y, r(t)\right)\Lambda''\left(y, r(t')\right)\, i\hat{h}(t')\right\rangle \\
&\equiv \alpha\Delta\frac{\delta}{\delta Y(t')}\langle s(t)\Lambda'(y, r(t))\rangle\Big|_{Y=0}.
\end{aligned} \tag{A.22}$$

In addition, from (A.19), one can derive an ordinary differential equation for the magnetization

$$\dot{m}(t) = -\lambda m(t) - \mu(t). \tag{A.23}$$

The brackets in the previous equations denote, at the same time, the average over the label y, the process $s(t)$, as well as the average over the noise $\xi(t)$ and both h_0 and $h(0)$, whose probability distributions are given by $P(h(0))$ and $P_0(h_0)$, respectively. In other words, one has a set of kernels, such as $M_R(t,t')$ and $M_C(t,t')$, that can be obtained as an average over the stochastic process for $h(t)$ and therefore must be computed self-consistently.

Finally, equation (A.18) gives rise to equation (20) of the main text while equation (A.19) gives rise to the equation for the evolution of the magnetization. Note that the norm of the weight vector $\mathbf{w}(t)$ can also be computed by sampling the stochastic process

$$\begin{aligned}
\dot{\mathrm{w}}(t) &= -\tilde{\lambda}(t)\mathrm{w}(t) + \int_0^t \mathrm{d}t' M_R(t,t')(\mathrm{w}(t') - m(t')h_0) + \xi(t) + h_0(\hat{\lambda}(t)m(t) - \mu(t)), \\
P(\mathrm{w}_0) &= \frac{1}{\sqrt{2\pi R}}\,\mathrm{e}^{-\mathrm{w}_0^2/(2R)},
\end{aligned} \tag{A.24}$$

J. Stat. Mech. (2021) 124008

from which one gets

$$C(t,t') = \langle \mathrm{w}(t)^2 \rangle. \tag{A.25}$$

A.3. Numerical solution of DMFT equations

The algorithm to solve the DMFT equations that are summed up in equation (A.20) is the most natural one. It can be understood in the following way. The outcome of the DMFT is the computation of the kernels and functions appearing in it, namely $m(t)$, $M_C(t,t')$ and so on. They are determined as averages over the stochastic process that is defined through them. Therefore, one needs to solve the system of equations in a self-consistent way. The straightforward way to do that is to proceed by iterations:

(a) We start from a random guess of the kernels that we use to sample the stochastic process (A.20) several times;
(b) We compute the averages over these multiple realizations to obtain the updates of the auxiliary functions (A.21) and kernels (A.22), along with the magnetization (A.23);
(c) We use these new guesses to sample multiple realizations of the stochastic process again;
(d) We repeat steps (b) and (c) until the kernels reach a fixed point.

As in all iterative solutions of fixed-point equations, it is natural to introduce some damping in the update of the kernels to avoid wild oscillations. Note that the DMFT fixed-point equations are deterministic, hence at the given initial condition the solution is unique. Indeed, the kernels computed by DMFT are causal and a simple integration scheme of the equations is just extending them progressively in time starting from their initial value, which is completely deterministic given the initial condition for the stochastic process. This procedure has been first implemented in [34, 39] and recently developed further in other applications [40, 41]. However, DMFT has a long tradition in condensed matter physics [20], where more involved algorithms have been developed.

Appendix B. Generalization error

The generalization error at any time step is defined as the fraction of mislabeled instances:

$$\varepsilon_{\mathrm{gen}}(t) \equiv \frac{1}{4}\mathbb{E}_{\mathbf{X},\mathbf{y},\mathbf{x}_{\mathrm{new}},y_{\mathrm{new}}}\left[\left(y_{\mathrm{new}} - \hat{y}_{\mathrm{new}}\left(\mathbf{w}(t)\right)\right)^2\right], \tag{B.1}$$

where $\{\mathbf{X},\mathbf{y}\}$ is the training set, $\mathbf{x}_{\mathrm{new}}$ is an unseen data point and $\hat{y}_{\mathrm{new}}$ is the estimator for the new label y_{new}. The dependence on the training set here is hidden in the weight vector $\mathbf{w}(t) = \mathbf{w}(t,\mathbf{X},\mathbf{y})$.

J. Stat. Mech. (2021) 124008

B.1. Perceptron with linear activation function

In this case, the estimator for a new label is $\hat{y}_{\rm new}\left(\mathbf{w}(t)\right) = \text{sign}\left(\mathbf{w}(t)^\top \mathbf{x}_{\rm new}\right)$. The generalization error in the infinite dimensional limit $d \to \infty$ has been computed in [28] and reads

$$\varepsilon_{\rm gen}(t) = \frac{1}{2}\,\text{erfc}\left(\frac{m(t)}{\sqrt{2\Delta\, C(t,t)}}\right). \tag{B.2}$$

B.2. Perceptron with door activation function

In this case, the estimator for a new label is $\hat{y}_{\rm new}\left(\mathbf{w}(t)\right) = \text{sign}\left(\frac{1}{d}(\mathbf{w}(t)^\top \mathbf{x}_{\rm new})^2 - L^2\right)$. From equation (B.1), we have that

$$\varepsilon_{\rm gen}(t) = \frac{1}{2}\left(1 - \mathbb{E}_{\mathbf{X},\mathbf{y},\mathbf{x}_{\rm new},y_{\rm new}}\left[y_{\rm new}\cdot \hat{y}_{\rm new}(\mathbf{w}(t))\right]\right). \tag{B.3}$$

We consider the second term of (B.3)

$$\mathbb{E}_{\mathbf{X},\mathbf{y},\mathbf{x}_{\rm new},y_{\rm new}}\left[y_{\rm new}\cdot \hat{y}_{\rm new}(\mathbf{w}(t))\right] = \mathbb{E}_{\mathbf{X},\mathbf{y},\mathbf{x}_{\rm new}}\left[\text{sign}\left(\frac{y_{\rm new}}{d}(\mathbf{w}(t)^\top \mathbf{x}_{\rm new})^2 - y_{\rm new}L^2\right)\right]. \tag{B.4}$$

In the high dimensional limit, the overlap between weight vector and data point at each time step concentrates

$$\frac{\mathbf{w}(t)^\top \mathbf{x}_{\rm new}}{\sqrt{d}} = \frac{\mathbf{w}(t)^\top}{\sqrt{d}}\left(c_{\rm new}\frac{\mathbf{v}^*}{\sqrt{d}} + \sqrt{\Delta}\,\mathbf{z}_{\rm new}\right) \xrightarrow[d\to\infty]{} c_{\rm new}\, m(t) + \sqrt{\Delta C(t,t)}\, z, \tag{B.5}$$

where $z \sim \mathcal{N}(0,1)$. Therefore, we obtain

$$\begin{aligned}
&\mathbb{E}_{\mathbf{X},\mathbf{y},\mathbf{x}_{\rm new},y_{\rm new}}\left[y_{\rm new}\cdot \hat{y}_{\rm new}(\mathbf{w}(t))\right] \\
&\quad\simeq \mathbb{E}_{c_{\rm new},z,y_{\rm new}}\left[\text{sign}\left(y_{\rm new}\left(c_{\rm new}\, m(t) + \sqrt{\Delta C(t,t)}\, z\right)^2 - y_{\rm new}L^2\right)\right] \\
&\quad= \mathbb{P}\left(y_{\rm new}\left(c_{\rm new}\, m(t) + \sqrt{\Delta C(t,t)}\, z\right)^2 \geqslant y_{\rm new}L^2\right) \\
&\qquad- \mathbb{P}\left(y_{\rm new}\left(c_{\rm new}\, m(t) + \sqrt{\Delta C(t,t)}\, z\right)^2 < y_{\rm new}L^2\right)
\end{aligned} \tag{B.6}$$

and the generalization error in the infinite dimensional limit $d \to \infty$ is

$$\begin{aligned}
\varepsilon_{\rm gen}(t) = (1-\rho)\text{erfc}\left(\frac{L}{\sqrt{2\Delta C(t,t)}}\right) + \frac{\rho}{2}\left(\text{erf}\left(\frac{L - m(t)}{\sqrt{2\Delta C(t,t)}}\right)\right. \\
\left. + \text{erf}\left(\frac{L + m(t)}{\sqrt{2\Delta C(t,t)}}\right)\right).
\end{aligned} \tag{B.7}$$

J. Stat. Mech. (2021) 124008

Appendix C. Oracle error

We call *oracle error* the classification error made by an ideal oracle that has access to the vector $\mathbf{v}^*$ that characterizes the centers of the clusters in the two models under consideration (see section 2 in the main text). We define the oracle's estimator $\hat{y}^O_{\text{new}}$ given a new data point $\mathbf{x}_{\text{new}}$ as

$$\hat{y}^O_{\text{new}} = \arg\max_{\tilde{y}_{\text{new}}} \mathrm{p}\left(\tilde{y}_{\text{new}}|\mathbf{x}_{\text{new}}\right), \tag{C.1}$$

where the prior over the label $\tilde{y}_{\text{new}}$ and the coefficient $\tilde{c}_{\text{new}}$ along with the channel distribution

$$\mathrm{p}\left(\mathbf{x}_{\text{new}}|\tilde{c}_{\text{new}}\right) \propto \exp\left[-\frac{1}{2\Delta}\|\mathbf{x}_{\text{new}} - \frac{\tilde{c}_{\text{new}}}{\sqrt{d}}\mathbf{v}^*\|_2^2\right] \tag{C.2}$$

are known. We can rewrite the probability in equation (C.1) as

$$\begin{aligned}
\mathrm{p}\left(\tilde{y}_{\text{new}}|\mathbf{x}_{\text{new}}\right) &\propto \sum_{\tilde{c}_{\text{new}}=0,\pm1} \mathrm{p}\left(\tilde{y}_{\text{new}}, \tilde{c}_{\text{new}}\right) \mathrm{p}\left(\mathbf{x}_{\text{new}}|\tilde{c}_{\text{new}}\right) \\
&= (1-\rho)\delta(\tilde{y}_{\text{new}}+1)\mathrm{e}^{-\frac{1}{2\Delta}\|\mathbf{x}_{\text{new}}\|_2^2} + \frac{\rho}{2}\delta(\tilde{y}_{\text{new}}-1)\left(\mathrm{e}^{-\frac{1}{2\Delta}\|\mathbf{x}_{\text{new}}-\frac{1}{\sqrt{d}}\mathbf{v}^*\|_2^2} + \mathrm{e}^{-\frac{1}{2\Delta}\|\mathbf{x}_{\text{new}}+\frac{1}{\sqrt{d}}\mathbf{v}^*\|_2^2}\right) \\
&= \mathrm{e}^{-\frac{1}{2\Delta}\|\mathbf{x}_{\text{new}}\|_2^2}\left[(1-\rho)\delta(\tilde{y}_{\text{new}}+1) + \rho\delta(\tilde{y}_{\text{new}}-1)\mathrm{e}^{-\frac{1}{2\Delta}}\cosh\left(\frac{1}{\Delta\sqrt{d}}\mathbf{x}_{\text{new}}^\top\mathbf{v}^*\right)\right].
\end{aligned} \tag{C.3}$$

The oracle error is then

$$\varepsilon^O_{\text{gen}} = \mathbb{P}\left(\hat{y}^O_{\text{new}} \neq y_{\text{new}}\right) = (1-\rho)\,\mathbb{P}\left(\hat{y}^O_{\text{new}} = 1|y_{\text{new}} = -1\right) + \rho\,\mathbb{P}\left(\hat{y}^O_{\text{new}} = -1|y_{\text{new}} = 1\right). \tag{C.4}$$

We can compute the two terms in the above equation separately

$$\begin{aligned}
\mathbb{P}\left(\hat{y}^O_{\text{new}} = 1|y_{\text{new}} = -1\right) &= \mathbb{P}\left(\rho\,\mathrm{e}^{-\frac{1}{2\Delta}}\cosh\left(\frac{1}{\sqrt{\Delta d}}\mathbf{z}_{\text{new}}^\top\mathbf{v}^*\right) > 1-\rho\right) \\
&= \mathbb{P}\left(\rho\,\mathrm{e}^{-\frac{1}{2\Delta}}\cosh\left(\frac{\zeta_{\text{new}}}{\sqrt{\Delta}}\right) > 1-\rho\right) \\
&= \mathrm{erfc}\left(\sqrt{\frac{\Delta}{2}}\left|\operatorname{arccosh}\left(\frac{(1-\rho)}{\rho}\,\mathrm{e}^{1/2\Delta}\right)\right|\right),
\end{aligned} \tag{C.5}$$

J. Stat. Mech. (2021) 124008

and

$$\begin{aligned}\mathbb{P}\left(\hat{y}^{O}_{\rm new}=-1|y_{\rm new}=1\right)&=\mathbb{P}\left(1-\rho>\rho\,{\rm e}^{-\frac{1}{2\Delta}}\,\cosh\left(\frac{c_{\rm new}}{\Delta}+\frac{1}{\sqrt{\Delta d}}\mathbf{z}_{\rm new}^{\top}\mathbf{v}^{*}\right)\right)\\&=\mathbb{P}\left(1-\rho>\rho\,{\rm e}^{-\frac{1}{2\Delta}}\,\cosh\left(\frac{c_{\rm new}}{\Delta}+\frac{\zeta_{\rm new}}{\sqrt{\Delta}}\right)\right)\\&=\frac{1}{2}\left[{\rm erf}\left(\frac{\Delta\left|{\rm arccosh}\left(\frac{(1-\rho)}{\rho}\,{\rm e}^{1/2\Delta}\right)\right|+1}{\sqrt{2\Delta}}\right)\right.\\&\quad\left.+\,{\rm erf}\left(\frac{\Delta\left|{\rm arccosh}\left(\frac{(1-\rho)}{\rho}\,{\rm e}^{1/2\Delta}\right)\right|-1}{\sqrt{2\Delta}}\right)\right],\end{aligned}\tag{C.6}$$

where $\mathbf{z}_{\rm new}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)$, $\zeta_{\rm new}\sim\mathcal{N}(0,1)$, and $c_{\rm new}=\pm1$ with probability 1/2.

Finally, we obtain that the oracle error is

$$\begin{aligned}\varepsilon^{BO}_{\rm gen}&=(1-\rho){\rm erfc}\left(\sqrt{\frac{\Delta}{2}}\left|{\rm arccosh}\left(\frac{(1-\rho)}{\rho}\,{\rm e}^{1/2\Delta}\right)\right|\right)\\&\quad+\frac{\rho}{2}\left[{\rm erf}\left(\frac{\Delta\left|{\rm arccosh}\left(\frac{(1-\rho)}{\rho}\,{\rm e}^{1/2\Delta}\right)\right|+1}{\sqrt{2\Delta}}\right)+{\rm erf}\left(\frac{\Delta\left|{\rm arccosh}\left(\frac{(1-\rho)}{\rho}\,{\rm e}^{1/2\Delta}\right)\right|-1}{\sqrt{2\Delta}}\right)\right].\end{aligned}\tag{C.7}$$

References

[1] Safran I and Shamir O 2017 Spurious local minima are common in two-layer relu neural networks (arXiv:1712.08968)

[2] Liu S, Papailiopoulos D and Achlioptas D 2019 Bad global minima exist and SGD can reach them (arXiv:1906.02613)

[3] Bös S and Opper M 1997 Dynamics of training *Advances in Neural Information Processing Systems* pp 141–7

[4] Saxe A M, McClelland J L and Ganguli S 2013 Exact solutions to the nonlinear dynamics of learning in deep linear neural networks (arXiv:1312.6120)

[5] Saad D and Solla S A 1995 Exact solution for on-line learning in multilayer neural networks *Phys. Rev. Lett.* **74** 4337

[6] Saad D and Solla S A 1995 On-line learning in soft committee machines *Phys. Rev.* E **52** 4225

[7] Saad D 2009 *On-Line Learning in Neural Networks* vol 17 (Cambridge: Cambridge University Press)

[8] Goldt S, Advani M, Saxe A M, Krzakala F and Zdeborová L 2019 Dynamics of stochastic gradient descent for two-layer neural networks in the teacher–student setup *Advances in Neural Information Processing Systems* pp 6979–89

[9] Goldt S, Mézard M, Krzakala F and Zdeborová L 2019 Modelling the influence of data structure on learning in neural networks (arXiv:1909.11500)

[10] Rotskoff G M and Vanden-Eijnden E 2018 Neural networks as interacting particle systems: asymptotic convexity of the loss landscape and universal scaling of the approximation error (arXiv:1805.00915)

[11] Song M, Montanari A and Nguyen P-M 2018 A mean field view of the landscape of two-layer neural networks *Proc. Natl Acad. Sci.* **115** E7665–71

J. Stat. Mech. (2021) 124008

[12] Chizat L and Bach F 2018 On the global convergence of gradient descent for over-parameterized models using optimal transport *Advances in Neural Information Processing Systems* pp 3036–46
[13] Poole B, Lahiri S, Raghu M, Sohl-Dickstein J and Ganguli S 2016 Exponential expressivity in deep neural networks through transient chaos *Advances in Neural Information Processing Systems* vol 29 ed D D Lee, M Sugiyama, U V Luxburg, I Guyon and R Garnett (New York: Curran Associates) pp 3360–8
[14] Schoenholz S S, Gilmer J, Ganguli S and Sohl-Dickstein J 2017 Deep information propagation (arXiv:1611.01232)
[15] Yang G, Pennington J, Rao V, Sohl-Dickstein J and Schoenholz S S 2019 A mean field theory of batch normalization *Int. Conf. on Learning Representations*
[16] Song M, Misiakiewicz T and Montanari A 2019 Mean-field theory of two-layers neural networks: dimension-free bounds and Kernel limit *Conf. on Learning Theory* pp 2388–464
[17] Gilboa D, Chang B, Chen M, Yang G, Schoenholz S S, Chi E H and Pennington J 2019 Dynamical isometry and a mean field theory of LSTMs and GRUs (arXiv:1901.08987)
[18] Novak R, Xiao L, Bahri Y, Lee J, Yang G, Abolafia D A, Pennington J and Sohl-dickstein J 2019 Bayesian deep convolutional networks with many channels are Gaussian processes *Int. Conf. on Learning Representations*
[19] Mézard M, Parisi G and Virasoro M A 1987 *Spin Glass Theory and Beyond* (Singapore: World Scientific)
[20] Antoine G, Gabriel K, Werner K and Rozenberg M J 1996 Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions *Rev. Mod. Phys.* **68** 13
[21] Parisi G, Urbani P and Zamponi F 2020 *Theory of Simple Glasses: Exact Solutions in Infinite Dimensions* (Cambridge: Cambridge University Press)
[22] Gabrié M 2020 Mean-field inference methods for neural networks *J. Phys. A: Math. Theor.* **53** 223002
[23] Ben Arous G *et al* 1997 Symmetric Langevin spin glass dynamics *Ann. Probab.* **25** 1367–422
[24] Mannelli S S, Biroli G, Cammarota C, Krzakala F, Urbani P and Zdeborová L 2020 Marvels and pitfalls of the Langevin algorithm in noisy high-dimensional inference *Phys. Rev.* X **10** 011057
[25] Mannelli S S, Krzakala F, Urbani P and Zdeborova L 2019 Passed & spurious: descent algorithms and local minima in spiked matrix-tensor models *Int. Conf. on Machine Learning* pp 4333–42
[26] Franz S, Parisi G, Sevelev M, Urbani P and Zamponi F 2017 Universality of the SAT-UNSAT (jamming) threshold in non-convex continuous constraint satisfaction problems *SciPost Phys.* **2** 019
[27] Franz S, Hwang S and Urbani P 2019 Jamming in multilayer supervised learning models *Phys. Rev. Lett.* **123** 160602
[28] Mignacco F, Krzakala F, Lu Y M and Zdeborová L 2020 The role of regularization in classification of high-dimensional noisy Gaussian mixture (arXiv:2002.11544)
[29] Rosset S, Zhu J and Hastie T J 2004 Margin maximizing loss functions *Advances in Neural Information Processing Systems* pp 1237–44
[30] Deng Z, Kammoun A and Thrampoulidis C 2019 A model of double descent for high-dimensional binary linear classification (arXiv:1911.05822)
[31] Seung H S, Opper M and Sompolinsky H 1992 Query by committee *Proc. 5th Annual Workshop on Computational Learning Theory* pp 287–94
[32] Agoritsas E, Biroli G, Urbani P and Zamponi F 2018 Out-of-equilibrium dynamical mean-field equations for the perceptron model *J. Phys. A: Math. Theor.* **51** 085002
[33] de Dominicis C 1976 Technics of field renormalization and dynamics of critical phenomena *J. Phys. Colloq.* **1** C1247
[34] Eissfeller H and Opper M 1992 New method for studying the dynamics of disordered spin systems without finite-size effects *Phys. Rev. Lett.* **68** 2094
[35] Soudry D, Hoffer E, Nacson M S, Gunasekar S and Nathan S 2018 The implicit bias of gradient descent on separable data *J. Mach. Learn. Res.* **19** 2822–78
[36] Kurchan J 2002 Supersymmetry, replica and dynamic treatments of disordered systems: a parallel presentation (arXiv:cond-mat/0209399)
[37] Kurchan J 1992 Supersymmetry in spin glass dynamics *J. Physique* I **2** 1333–52
[38] Zinn-Justin J 1996 *Quantum Field Theory and Critical Phenomena* (Oxford: Clarendon)
[39] Eissfeller H and Opper M 1994 Mean-field Monte Carlo approach to the Sherrington–Kirkpatrick model with asymmetric couplings *Phys. Rev.* E **50** 709
[40] Roy F, Biroli G, Bunin G and Cammarota C 2019 Numerical implementation of dynamical mean field theory for disordered systems: application to the Lotka–Volterra model of ecosystems *J. Phys. A: Math. Theor.* **52** 484001
[41] Manacorda A, Schehr G and Zamponi F 2020 Numerical solution of the dynamical mean field theory of infinite-dimensional equilibrium liquids *J. Chem. Phys.* **152** 164506

J. Stat. Mech. (2021) 124008