Double Trouble in Double Descent: Bias and Variance(s) in the Lazy Regime

Stéphane d'Ascoli [*1] **Maria Refinetti** [*1] **Giulio Biroli** [1] **Florent Krzakala** [1]

Abstract

Deep neural networks can achieve remarkable generalization performances while interpolating the training data; rather than the U-curve emblematic of the bias-variance trade-off, their test error often follows a "double descent" curve — a mark of the beneficial role of overparametrization. In this work, we develop a quantitative theory for this phenomenon in the context of high-dimensional random features regression. We obtain a precise asymptotic expression for the bias-variance decomposition of the test error, and show that the bias displays a phase transition at the interpolation threshold, beyond it which it remains constant. We disentangle the variances stemming from the sampling of the dataset, from the additive noise corrupting the labels, and from the initialization of the weights. Following up on (Geiger et al., 2019a), we demonstrate that the latter two contributions are the crux of the double descent: they lead to the overfitting peak at the interpolation threshold and to the decay of the test error upon overparametrization. We quantify how they are suppressed by averaging the outputs of independently initialized estimators, and compare this ensembling procedure with overparametrization and regularization. Finally, we present numerical experiments on a standard deep learning setup to show that our results are relevant to the lazy regime of deep neural networks.

1. Introduction

Deep neural networks have achieved breakthroughs in a plethora of contexts, such as image classification (Krizhevsky et al., 2012; LeCun et al., 2015), speech recognition (Hinton et al., 2012), and automatic translation (Sutskever et al., 2014). Yet, theory lags far behind practice, and the key reasons underpinning the success of DNNs remain to be clarified.

One of the main puzzles is to understand the excellent generalization performance of heavily overparametrized deep neural networks able to fit random labels (Zhang et al., 2016). Such *interpolating* estimators—that can reach zero training error— have attracted a growing amount of theoretical attention in the last few years, see e.g. (Advani & Saxe, 2017; Belkin et al., 2018; Neal et al., 2018; Hastie et al., 2019; Mei & Montanari, 2019). Indeed, classical learning theory suggests that generalization should first improve then worsen when increasing model complexity, following a U-shape curve characteristic of the bias-variance trade-off. Instead, deep neural networks (Neyshabur et al., 2014; Spigler et al., 2018; Nakkiran et al., 2019) as well as other machine learning models (Belkin et al., 2018), follow a different curve, coined *double descent*.

This curve displays two regimes : the *classical* U-curve is superseded at high complexity by a *modern* interpolating regime where the test error decreases monotonically with overparametrization (Breiman, 1995). Between these two regimes, i.e. at the *interpolation threshold* where training error vanishes, a *peak* occurs in absence of regularization, sometimes called the *jamming* peak due to similarities with a well-studied phenomenon in the Statistical Physics litterature (Opper & Kinzel, 1996; Engel & Van den Broeck, 2001; Franz & Parisi, 2016; Spigler et al., 2018; Krzakala & Kurchan, 2007; Zdeborová & Krzakala, 2007). The reasons behind the performance of deep neural networks in the overparametrized regime are still poorly understood, even though some mechanisms are known to play an important role, such as the implicit regularization of stochastic gradient descent which allows to converge to the minimum norm solution, and the convergence to mean-field limits (Rosset et al., 2004; Advani & Saxe, 2017; Chizat et al., 2019; Arora et al., 2019a; Mei et al., 2019).

Here we present a detailed investigation of the double descent phenomenon, and its theoretical explanation in terms of bias and variance in the so-called *lazy* regime (Chizat et al., 2019). This theoretically appealing scenario, where the weights stay close to their initial value during train-

[*]Equal contribution [1]Laboratoire de Physique de l'Ecole normale supérieure, ENS, Université PSL, CNRS, Sorbonne Université, Université de Paris, F-75005 Paris, France. Correspondence to: Stephane d'Ascoli <stephane.dascoli@gmail.com>, Maria Refinetti <mariaref@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

ing, is called *lazy learning* as opposed to *feature learning* where the weights change enough to learn relevant features (Chizat et al., 2019; Woodworth et al., 2019; Geiger et al., 2019b). Although replacing learnt features by random features may appear as a crude simplification, empirical results show that the loss in performance can be rather small in some cases (Arora et al., 2019b;a). A burst of recent papers showed that in this regime, neural networks behave like kernel methods (Daniely et al., 2016; Lee et al., 2017; Jacot et al., 2018) or equivalently random projection methods (Rahimi & Recht, 2008; Chizat et al., 2019; Arora et al., 2019a). This mapping makes the training analytically tractable, allowing, for example, to prove convergence to zero error solutions in overparametrized settings.

Optimization plays an important role in neural networks by inducing implicit regularization (Neyshabur et al., 2017) and fluctuations of the learnt estimator (Geiger et al., 2019a). Disentangling the variance stemming from the randomness of the optimization process from that the variance due to the randomness of the dataset is a crucial step towards a unified picture, as suggested in (Neal et al., 2018). In this paper, we address this issue and attempt to reconcile the behavior of bias and variance with the double descent phenomenon by providing a precise and quantitative theory in the *lazy* regime.

Contributions We focus on an analytically solvable model of random features (RF), introduced by (Rahimi & Recht, 2008), that can be viewed either as a randomized approximation to kernel ridge regression, or as a two-layer neural network whose first layer contains fixed random weights. The latter provides a simple model for lazy learning. Indeed, suppose that a neural network learns a function $f_\theta(\mathbf{x})$ that relates labels (or responses) y_i to inputs $\mathbf{x_i}$ with, $i = 1, \dots, N$ via a set of weights θ. The lazy regime is defined as the setting where the model can be linearized around the initial conditions θ_0. Assuming that the initialization is such that f_{θ_0} 0[1], one obtains:

$$f_\theta(\mathbf{x}) \quad r_\theta f_\theta(\mathbf{x}) j_{\theta=\theta_0} \quad (\theta \quad \theta_0) . \tag{1}$$

In other words, the lazy regime corresponds to a linear fitting problem with a random feature vector $r_\theta f_\theta(\mathbf{x}) j_{\theta=\theta_0}$.

In this setting our contributions are:

- We demonstrate how to disentangle quantitatively the contributions to the test error of the bias and the various sources of variance of the estimator, stemming from the sampling of the dataset, from the additive noise corrupting the labels, and from the initialization of the random feature vectors.
- We give a sharp asymptotic formula for the effect of *ensembling* (averaging the predictions of indepently initialized estimators) on these various terms. We show in particular how the over-fitting peak at the interpolation threshold can be attenuated by ensembling, as observed in real neural networks (Geiger et al., 2019a). We also compare the effect of ensembling, overparametrizing and optimally regularizing.
- Several conclusions stem from the above analysis. First, the over-fitting near the interpolation threshold is entirely due to the variances due to the additive noise in the ground truth and the initialization of the random features. Second, the data sampling variance and the bias both display a phase transition at the interpolation threshold, and remain constant in the overparametrized regime. Hence, the benefit of ensembling and overparametrization beyond the interpolation threshold is solely due to a reduction of the noise and initialization variances.

Finally, we present numerical results on a classic deep learning scenario in the lazy learning regime to show that our findings, obtained for simple random features and i.i.d. data, are relevant to realistic setups involving correlated random features and realistic data.

The analytical results we present are obtained using a heuristic method from Statistical Physics called the Replica Method (Mézard et al., 1987), which despite being non-rigorous has shown its remarkable efficacy in many machine learning problems (Seung et al., 1992; Engel & Van den Broeck, 2001; Advani et al., 2013; Zdeborová & Krzakala, 2016) and random matrix topics, see e.g. (Livan et al., 2018; Tarquini et al., 2016; Aggarwal et al., 2018). While it is an open problem to provide a rigorous proof of our computations, we check through numerical simulations that our asymptotic predictions are extremely accurate at moderately small sizes.

Related work On the empirical side, the authors of (Geiger et al., 2019a) carried out a series of experiments in order to shed light on the generalization properties of neural networks. The current work is partly inspired by their observations and arguments about the role of the variance due to the random initialization of the weights in the double-descent curve. Another related work is (Neal et al., 2018), which disentangles the various sources of variance in the process of training deep neural networks.

On the theoretical side, our paper builds on the results of (Mei & Montanari, 2019), which provide an analytic expression of the test error of the RF model in the high-dimensional limit where the number of random features, the dimension of the input data and the number of data points are sent to infinity with their relative ratios fixed. The double descent was also studied analytically for various types of linear

[1] One can alteratively define the estimator as f_θ f_{θ_0} (Chizat et al., 2019).

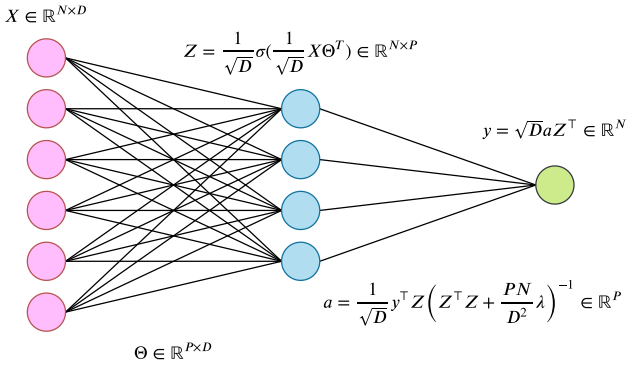

Figure 1. Illustration of a RF network. The first layer weights are fixed and initialized as i.i.d. centered Gaussian variables of unit variance. The second layer weights are trained via ridge regression.

models, both for regression (Advani & Saxe, 2017; Hastie et al., 2019; Nakkiran, 2019; Belkin et al., 2019; d'Ascoli et al., 2020) and classification (Deng et al., 2019; Kini & Thrampoulidis, 2020; Gerace et al., 2020).

An example of a practical method that uses ensembling in kernel methods is detailed in (Drucker et al., 1994). Note that this work performs an average over the sampling of the random feature vectors in contrast to (Zhang et al., 2013) where the average is taken over the sampling of the data set.

Reproducibility The codes necessary to reproduce the results presented in this paper and obtain new ones are given at:

`https://github.com/mariaref/Random_Features.git`

2. Model

This work is centered around the RF model first introduced in (Rahimi & Recht, 2008). Although simpler settings such as linear regression display the double descent phenomenology (Hastie et al., 2019), this model is more appealing in several ways. First, the presence of two layers allows to freely disentangle the dimensionality of the input data from the number of parameters of the model. Second, it closely relates to the lazy learning, as described above. Third, and most importantly for our specific study, the randomness of the first layer weights enables to study the impact of ensembling.

We consider a regression task involving a two-layer neural network whose first layer contains fixed random weights[2] (see figure 1):

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{P} \boldsymbol{a}_i \sigma\left(\frac{\langle \boldsymbol{\theta}_i, \boldsymbol{x}\rangle}{\sqrt{D}}\right). \tag{2}$$

In the above, $\boldsymbol{\theta}_i$ is the i^{th} *random feature*, i.e the i^{th} column of the random feature matrix $\boldsymbol{\Theta} \in \mathbb{R}^{P\times D}$ whose elements are drawn i.i.d from $\mathcal{N}(0,1)$. σ is a pointwise activation function, which we will take to be $ReLU: x \mapsto \max(0,x)$.

The training data is collected in a matrix $\boldsymbol{X} \in \mathbb{R}^{N\times D}$ whose elements are drawn i.i.d from $\mathcal{N}(0,1)$. We assume that the labels are given by a linear ground truth corrupted by some additive Gaussian noise:

$$y_\mu = \langle \boldsymbol{\beta}, \boldsymbol{X}_\mu\rangle + \epsilon_\mu, \quad \|\boldsymbol{\beta}\| = F, \quad \epsilon_\mu \sim \mathcal{N}(0,\tau), \tag{3}$$
$$\mathrm{SNR} = F/\tau.$$

The generalization to non-linear functions can also be performed as in (Mei & Montanari, 2019).

The second layer weights, i.e the elements of $\boldsymbol{a}$, are calculated by the means of ridge regression:

$$L_{\mathrm{RF}}(\boldsymbol{a}) \equiv \frac{1}{N}\sum_{\mu=1}^{N}\left(y_\mu - \sum_{i=1}^{P} \boldsymbol{a}_i \sigma\left(\frac{\langle \boldsymbol{\theta}_i, \boldsymbol{X}_\mu\rangle}{\sqrt{D}}\right)\right)^2 + \frac{P\lambda}{D}\|\boldsymbol{a}\|_2^2,$$
$$\hat{\boldsymbol{a}} \equiv \arg\min_{\boldsymbol{a}\in\mathbb{R}^P} L_{\mathrm{RF}}(\boldsymbol{a}).$$

Note that as $P \to 1$, this is equivalent to kernel ridge regression with respect to the following kernel:

$$K(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}_{\boldsymbol{\theta}\sim P}\left[\sigma(\langle \boldsymbol{x}, \boldsymbol{\theta}\rangle/\sqrt{D})\sigma\left(\langle \boldsymbol{x}', \boldsymbol{\theta}\rangle/\sqrt{D}\right)\right],$$

where $P = \mathrm{Unif}\left(\mathbb{S}^{D-1}(\sqrt{D})\right)$.

The key quantity of interest is the *test error* of this model, defined as the mean square error evaluated on a fresh sample $\boldsymbol{x} \sim \mathcal{N}(0,1)$ corrupted by a new noise $\tilde{\epsilon}$:

$$R_{\mathrm{RF}} = \mathbb{E}_x\left[\left(\langle \boldsymbol{\beta}, \boldsymbol{x}\rangle + \tilde{\epsilon} - \hat{f}(\boldsymbol{x})\right)^2\right], \quad \tilde{\epsilon} \sim \mathcal{N}(0,\tilde{\tau}). \tag{4}$$

3. Analytical results

In this section, we present our main result, which is an analytical expression for all terms appearing in the decomposition of the test error in terms of its bias and variance components.

3.1. Decomposition of the test error

The risk function (4) can be decomposed into five terms:

$$\mathbb{E}_{\boldsymbol{\Theta},\boldsymbol{X},\boldsymbol{\epsilon}}[R_{\mathrm{RF}}] = E_{\mathrm{Noise}} + E_{\mathrm{Init}} + E_{\mathrm{Samp}} + E_{\mathrm{Bias}} + \tilde{\tau}^2. \tag{5}$$

[2] Note the closeness between the RF model and the "*hidden manifold model*" introduced in (Goldt et al., 2019). The task studied here can be seen as a linear regression task on a structured data set $\boldsymbol{Z} \in \mathbb{R}^P$, obtained by projecting the original *latent* features $\boldsymbol{X} \in \mathbb{R}^D$. The difference here is that the dimension of the latent space, denoted as D here, is sent to infinity together with the dimension of the ambient space.

The first three contribute to the variance, the fourth is the bias, and the final term $\tilde{\tau}^2$ is simply the error of an *oracle* predictor. It does not play any role and will be set to zero in the rest of the paper: the only reason it was included is to avoid confusion with E_{Noise} defined below.

Noise variance: The first term is the variance associated with the additive noise corrupting the labels of the dataset which is learnt, ε:

$$E_{\text{Noise}} = \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{X},\boldsymbol{\Theta}} \left[\mathop{\mathbb{E}}_{\varepsilon} \left[\hat{f}(\boldsymbol{x})^2 \right] \quad \left(\mathop{\mathbb{E}}_{\varepsilon} \left[\hat{f}(\boldsymbol{x}) \right] \right)^2 \right]. \tag{6}$$

Initialization variance: The second term encodes the fluctuations stemming from the random initialization of the random feature vectors, $\boldsymbol{\Theta}$:

$$E_{\text{Init}} = \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{X}} \left[\mathop{\mathbb{E}}_{\boldsymbol{\Theta}} \left[\mathop{\mathbb{E}}_{\varepsilon} \left[\hat{f}(\boldsymbol{x}) \right]^2 \right] \quad \mathop{\mathbb{E}}_{\boldsymbol{\Theta},\varepsilon} \left[\hat{f}(\boldsymbol{x}) \right]^2 \right].$$

Sampling variance: The third term measures the fluctuations due to the sampling of the training data, $\boldsymbol{X}$:

$$E_{\text{Samp}} = \mathop{\mathbb{E}}_{\boldsymbol{x}} \left[\mathop{\mathbb{E}}_{\boldsymbol{X}} \left[\mathop{\mathbb{E}}_{\boldsymbol{\Theta},\varepsilon} \left[\hat{f}(\boldsymbol{x}) \right]^2 \right] \quad \mathop{\mathbb{E}}_{\boldsymbol{X},\boldsymbol{\Theta},\varepsilon} \left[\hat{f}(\boldsymbol{x}) \right]^2 \right]. \tag{7}$$

Bias: Finally, the last term in (5) is the bias, i.e. the error that remains once all the sources of variance have been averaged out. It can be understood as the approximation error of our model and takes the form:

$$E_{\text{Bias}} = \mathop{\mathbb{E}}_{\boldsymbol{x}} \left[\left(h\boldsymbol{\beta}, \boldsymbol{x} i \quad \mathop{\mathbb{E}}_{\boldsymbol{X},\boldsymbol{\Theta},\varepsilon} \left[\hat{f}(\boldsymbol{x}) \right] \right)^2 \right]. \tag{8}$$

Note that this decomposition is sequential: we first remove the noise variance, then remove the initialization variance from the noise averaged predictor, and finally remove the residual sampling variance from the noise and initialization averaged predictor. This order was chosen to enable comparison with other bias-variance decompositions (Mei & Montanari, 2019; Neal et al., 2018).

Other sources of variance By performing ridge regression, we are missing out on two sources of variance which could be incurred by SGD dynamics. First, the noise in SGD creates an extra source of variance. Second, even noiseless GD would add an extra contribution to initialization variance. Indeed, in the overparametrized regime, the ridge regression problem is underdetermined: there is a frozen part of the estimator which cannot be learnt (Advani & Saxe, 2017). This part, which is set to zero in ridge regression, adds an extra (harmful) dependency on initialization in GD.

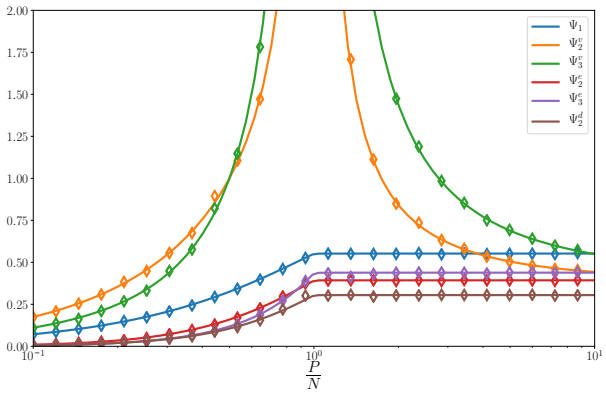

Figure 2. All the terms entering our analytical expressions for the decomposition of the error (Equations 10-15), as function of the overparametrization ratio P/N for $\lambda = 10^{-5}$ and $N/D = 1$. Numerical estimations obtained for a finite size $D = 200$ (diamonds) shows that the asymptotic predictions are extremely accurate even at moderate sizes.

3.2. Main Analytical result

Consider the high-dimensional limit where the input dimension D, the hidden layer dimension P (which is equal to the number of parameter in our model) and the number of training points N go to infinity with their ratios fixed:

$$N, P, D \to \infty, \quad \frac{P}{D} = O(1), \quad \frac{N}{D} = O(1). \tag{9}$$

We obtain the following result:

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\varepsilon,\boldsymbol{\Theta},\boldsymbol{X}} \left[h\boldsymbol{\beta}, \boldsymbol{x} i \hat{f}(\boldsymbol{x}) \right] = F^2 \Psi_1, \tag{10}$$

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{\Theta},\boldsymbol{X}} \left[\mathop{\mathbb{E}}_{\varepsilon} \left[\hat{f}(\boldsymbol{x}) \right]^2 \right] = F^2 \Psi_2^v, \tag{11}$$

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{\Theta},\boldsymbol{X}} \left[\mathop{\mathbb{E}}_{\varepsilon} \left[\hat{f}(\boldsymbol{x})^2 \right] \quad \mathop{\mathbb{E}}_{\varepsilon} \left[\hat{f}(\boldsymbol{x}) \right]^2 \right] = \tau^2 \Psi_3^v, \tag{12}$$

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{X}} \left[\mathop{\mathbb{E}}_{\varepsilon,\boldsymbol{\Theta}} \left[\hat{f}(\boldsymbol{x}) \right]^2 \right] = F^2 \Psi_2^e, \tag{13}$$

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{X}} \left[\mathop{\mathbb{E}}_{\varepsilon,\boldsymbol{\Theta}} \left[\hat{f}(\boldsymbol{x})^2 \right] \quad \mathop{\mathbb{E}}_{\varepsilon,\boldsymbol{\Theta}} \left[\hat{f}(\boldsymbol{x}) \right]^2 \right] = \tau^2 \Psi_3^e, \tag{14}$$

$$\mathop{\mathbb{E}}_{\boldsymbol{x}} \left[\mathbb{E}_{\varepsilon,\boldsymbol{\Theta},\boldsymbol{X}} \left[\hat{f}(\boldsymbol{x}) \right]^2 \right] = F^2 \Psi_2^d, \tag{15}$$

where the terms $\{\Psi_1, \Psi_2^v, \Psi_3^v, \Psi_2^e, \Psi_3^e, \Psi_2^d\}$ are computed following methods developed in Statistical Physics. The main steps of our procedure are as follows:

1. *Mapping to a random matrix theory problem.* The first step is to express the right-hand sides of Equations 10-15 as traces over random matrices. This is achieved by replacing our model with its asymptotically equivalent Gaussian covariate model (Mei & Montanari, 2019), in

which the non-linearity of the activation function is encoded as an extra noise term. This enables to take the expectation value with respect to the test sample $\boldsymbol{x}$.

2. *Mapping to a statistical physics model.* The random matrix theory problem resulting from the solution of ridge regression (4) involves inverse random matrices. In order to evaluate their expection value, we use the formula: $M_{ij}^{-1} = \lim_{n \to 0} \int \prod_{\alpha=1}^{n} \prod_{i=1}^{D} d\eta_i^\alpha \eta_i^1 \eta_j^1 \, e^{\frac{1}{2}\eta_i^\alpha M_{ij} \eta_j^\alpha}$, which is based on the Replica Trick (Mézard et al., 1987; Bun et al., 2016). The Gaussian integrals over $\boldsymbol{\varepsilon}, \boldsymbol{\Theta}, \boldsymbol{X}$ can then be straightforwardly performed and leads to a Statistical Physics model for the auxiliary variables η_i^α.
3. *Mean-Field Theory.* The model for the η_i^α variables can then be solved by introducing as order parameters the $n \times n$ overlap matrices $Q^{\alpha\beta} = \frac{1}{P}\sum_{i=1}^{P} \eta_i^\alpha \eta_i^\beta$ and using replica theory (Mézard et al., 1987), see the supplemental material (SM) for the detailed computation[3].

The full analytical expressions resulting from this procedure are deferred to the SM. The Ψ's may also be estimated numerically at finite size by evaluating the traces of the random matrices appearing in the Gaussian covariate model at the end of step 1. Figure 2 shows that results thus obtained are in excellent agreement with the asymptotic expressions even at moderate sizes, e.g. $D = 200$, proving the robustness of steps 2 and 3, which differ from the approach presented in (Mei & Montanari, 2019).

The indices v, e, d in $\{\Psi_1, \Psi_2^v, \Psi_3^v, \Psi_2^e, \Psi_3^e, \Psi_2^d\}$ stand for *vanilla*, *ensemble* and *divide and conquer*. The *vanilla* terms, which amount to a bias-variance decomposition with respect to noise in the labels, are sufficient to obtain the test error of a single RF model and were computed in (Mei & Montanari, 2019). The *ensemble* terms and *divide and conquer* terms, which are new, respectively allow to study initialization and sampling variance, and hence obtain the test error given by averaging the predictions of several different learners trained on the same dataset (ensembling) and on different splits of the dataset (divide and conquer), see section 5. Figure 2 shows that the *vanilla* terms exhibit a radically different behavior from the others: at vanishing regularization, they diverge at $P = N$ then decrease monotonically, whereas the others display a kink followed by a plateau. This behavior will be key to the following analysis.

[3]In order to obtain the asymptotic formulas for the Ψ's we need to compute (what are called in the Statistical Physics jargon) fluctuations around mean-field theory.

4. Analysis of Bias and Variances

The results of the previous section, allow to rewrite the decomposition of the test error as follows:

$$E_{\text{Noise}} = \tau^2 \Psi_3^v, \tag{16}$$

$$E_{\text{Init}} = F^2 (\Psi_2^v \quad \Psi_2^e), \tag{17}$$

$$E_{\text{Samp}} = F^2 \begin{pmatrix} \Psi_2^e & \Psi_2^d \end{pmatrix}, \tag{18}$$

$$E_{\text{Bias}} = F^2 \begin{pmatrix} 1 & 2\Psi_1 + \Psi_2^d \end{pmatrix}. \tag{19}$$

These contributions, together with the test error, are shown in figure 3 in the case of small (top) and large (bottom) regularization.

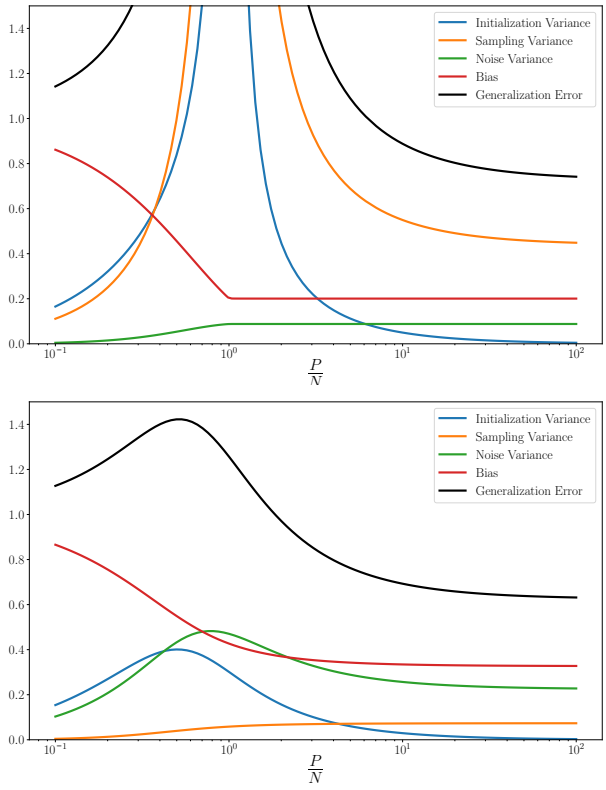

Figure 3. Decomposition of the test error into the bias and the various sources of variance as function of the overparametrization ratio P/N for $N/D = 1$, $\text{SNR} = F/\tau = 1$. Two values of the regularization constant are used: $\lambda = 10^{-5}$ (**top**) and $\lambda = 10^{-1}$ (**bottom**). Notice the contrasting behaviors at the interpolation threshold: the noise and initialization variances diverge then decrease monotonically whereas the sampling variance and the bias display a kink followed by a plateau. These singular behaviors are smoothed out by regularization.

Interpolation Threshold The peak at the interpolation threshold is completely due to noise and initialization variance, which both diverge at vanishing regularization. In contrast, the sampling variance and the bias remain finite and exhibit a phase transition at $P = N$, which is revealed

by a kink at vanishing regularization. Adding regularization smooths out these singular behaviours: it removes the divergence and irons out the kink.

Overparametrized regime In the overparametrized regime, the sampling variance and the bias do not vary substantially (they remain constant for vanishing regularization). The decrease of the test error is entirely due to the decrease of the noise and initalization variances for $P > N$. In the limit $P/N \to \infty$, the initialization variance vanishes, whereas there remains an irreducible noise variance.

In conclusion, we find that the *origin of the double descent curve lies in the behavior of noise and initialization variances*. The benefit of overparametrizing stems only from reducing these two contributions.

These results are qualitatively similar to the empirical decomposition of (Neal et al., 2018) for real neural networks. Our results differ however from those of (Mei & Montanari, 2019) where the authors relate the over-fitting peak occurring at $P = N$ to a divergence in both the variance and the bias terms. This is due to the fact the bias term, as defined in that paper, also includes the initialization variance[4]. When the two are disentangled, it becomes clear that it is only the latter which is responsible for the divergence: the bias is, in fact, well-behaved at $P = N$.

Discussion The phenomenology described above can be understood by noting that the model essentially performs linear regression, learning a vector $\boldsymbol{a} \in \mathbb{R}^P$ on a projected dataset $\boldsymbol{Z} \in \mathbb{R}^{N \times P}$ (the activations of the hidden nodes of the RF network). Since $\boldsymbol{a}$ is constrained to lie in the space spanned by $\boldsymbol{Z}$, which is of dimension $\min(N, P)$, the model gains expressivity when P increases while staying smaller than N.

At $P = N$, the problem becomes fully determined: the data is perfectly interpolated for vanishing λ. Two types of noise are overfit: (i) the *stochastic noise* corrupting the labels, yielding the divergence in noise variance, and (ii) the *deterministic noise*(Abu-Mostafa et al., 2012; d'Ascoli et al., 2020) stemming from the non-linearity of the activation function which cannot be captured, yielding the divergence in initialization variance. However, by further increasing P, the noise is spread over more and more random features and is effectively averaged out. Consequently, the test error decreases again as P increases.

When we make the problem deterministic by averaging out all sources of randomness, i.e. by considering the bias, we see that increasing P beyond N has no effect whatsoever. Indeed, the extra degrees of freedom, which lie in the null space of $\boldsymbol{Z}$, do not provide any extra expressivity: at vanishing regularization, they are killed by the pseudo-inverse to reach the minimum norm solution. For non-vanishing λ, a similar phenomenology is observed but the interpolation threshold is reached slightly after $P = N$ since the expressivity of the learner is lowered by regularization.

[4]For a given set of random features this is legitimate, but from the perspective of lazy learning the randomness in the features corresponds to the one due to initialization, which is an additional source of variance.

5. On the effect of ensembling

In order to further study the effect of the variances on the test error, we wish to study the impact of ensembling. In the lazy regime of deep neural networks, the initial values of the weights only affect the gradient at initialization, which corresponds to the vector of random features. Hence, we can study the effect of ensembling in the lazy regime by averaging the predictions of RF models with independently drawn random feature vectors.

Expression of the test error Consider a set of $K > 1$ RF networks whose first layer weights are drawn independently. These networks are trained independently on the *same* training set. In the analogy outlined above, they correspond to K independent inizializations of the neural network. At the end of training, one obtains K estimators $\hat{f}_{\boldsymbol{\Theta}_k}$ $(k = 1, ..., K)$. When a new sample $\boldsymbol{x}$ is presented to the system, the output is taken to be the average over the outputs of the K networks, as illustrated in figure 4. By expanding the square and taking the expectation with respect to the random initalizations, the test error can then be written as:

$$\begin{aligned}
\mathop{\mathbb{E}}_{f_{\boldsymbol{\Theta}^k}} [R_{\mathrm{RF}}] &= \mathop{\mathbb{E}}_{\substack{\boldsymbol{x} \\ f_{\boldsymbol{\Theta}^k}}} \left[\left(\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle - \frac{1}{K} \sum_k \hat{f}_{\boldsymbol{\Theta}_k}(\boldsymbol{x}) \right)^2 \right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}} \left[\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle^2 \right] - \frac{2}{K} \sum_{i=1}^{K} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x} \\ \boldsymbol{\Theta}_i}} \left[\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle \hat{f}_{\boldsymbol{\Theta}_i}(\boldsymbol{x}) \right] \\
&\quad + \frac{1}{K^2} \sum_{i,j=1}^{K} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x} \\ \boldsymbol{\Theta}_i, \boldsymbol{\Theta}_j}} \left[\hat{f}_{\boldsymbol{\Theta}_i}(\boldsymbol{x}) \hat{f}_{\boldsymbol{\Theta}_j}(\boldsymbol{x}) \right] . \qquad (20)
\end{aligned}$$

The key here is to isolate in the double sum the $K(K-1)$ *ensemble* terms $i \neq j$, which involve two different initalizations and yield $\mathbb{E}_{\boldsymbol{x}} \left[\mathbb{E}_{\boldsymbol{\Theta}} \left[\hat{f}_{\boldsymbol{\Theta}}(\boldsymbol{x}) \right]^2 \right]$, from the K *vanilla* terms which give $\mathbb{E}_{\boldsymbol{x}, \boldsymbol{\Theta}} \left[\hat{f}_{\boldsymbol{\Theta}}(\boldsymbol{x})^2 \right]$. This allows to express the test error in terms of the quantities defined in (10) to (15) and leads to the analytic formula for the test error valid for any $K \in \mathbb{N}$:

$$\begin{aligned}
\mathop{\mathbb{E}}_{f_{\boldsymbol{\Theta}^{(k)}}, \boldsymbol{X}, \boldsymbol{\varepsilon}} [R_{\mathrm{RF}}] &= F^2 (1 - 2\Psi_1^v) + \frac{1}{K} \left(F^2 \Psi_2^v + \tau^2 \Psi_3^v \right) \\
&\quad + \left(1 - \frac{1}{K} \right) \left(F^2 \Psi_2^c + \tau^2 \Psi_3^c \right) . \qquad (21)
\end{aligned}$$

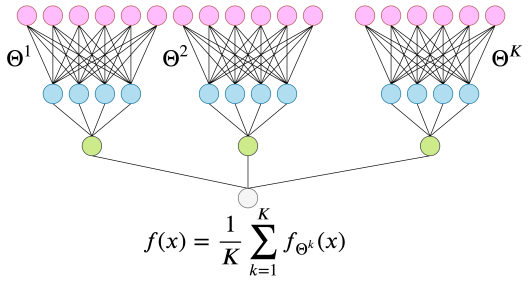

Figure 4. Illustration of the ensembling procedure over K RF networks trained on the same data but with different realizations of the first layer, $f\Theta_1, \ldots \Theta_K g$.

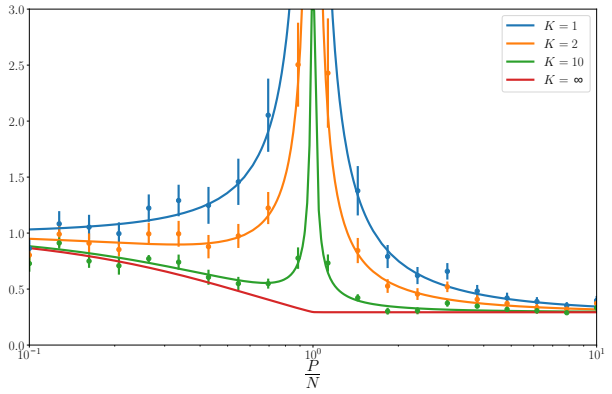

Figure 5. Test error when ensembling $K = 1, 2, 10$ differently initialized RF models as function of the overparametrization ratio P/N. We fixed $\lambda = 10^{-5}$, $N/D = 1$, SNR $= 1$. For comparison, we show the results of numerical simulations at finite $D = 200$: the vertical bars depict the standard deviation over 10 runs[6]. Note that our analytic expression 21 gives us access to the limit $N \to \infty$, where the divergence at $P = N$ is entirely suppressed.

We see that ensembling amounts to a linear interpolation between the *vanilla* terms Ψ_2^v, Ψ_3^v, for $K = 1$, and the *ensemble* terms Ψ_2^e, Ψ_3^e for $K \to \infty$.

The effect of ensembling on the double descent curve is shown in figure 5. As K increases, the overfitting peak at the interpolation threshold is diminished. This observation is very similar to the empirical findings of (Geiger et al., 2019a) in the context of real neural networks. Our analytic expression agrees with the numerical results obtained by training RF models, even at moderate size $D = 200$.

Note that a related procedure is the *divide and conquer* approach, where the dataset is partitioned into K splits of equal size and each one of the K differently initialized learners is trained on a distinct split. This approach and was studied for kernel learning in (Drucker et al., 1994), and is analyzed within our framework in the SM.

[6]The variability observed here was absent in figure 2 because we are considering the true RF model rather than the asymptotically equivalent Gaussian covariate model. This shows that most of this variability is caused by the finite-size deviation between the two models.

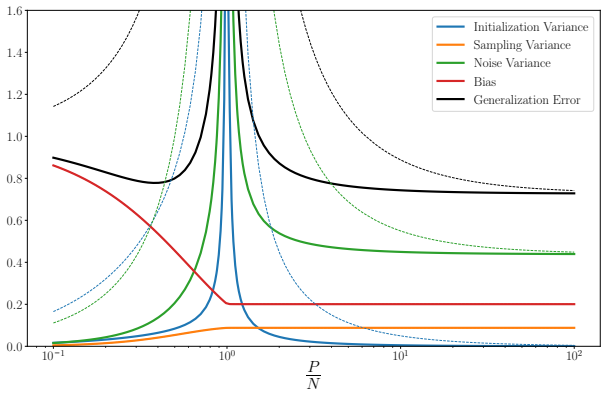

Figure 6. Decomposition of the test error into the bias and the various sources of variance as function of the overparametrization ratio P/N for $\lambda = 10^{-5}$, $N/D = 1$, SNR $= 1$. The thin dashed lines are taken from figure 3 (top) where we had $K = 1$; the thick solid lines show how ensembling at $K = 10$ suppressed the divergences of the noise and initialization variances.

Ensembling reduces the double trouble The bias-variance decomposition of the test error makes the suppression of the divergence explicit. The bias and variances contribution read for the averaged estimator:

$$E_{\text{Noise}} = \tau^2 \left(\Psi_3^e + \frac{1}{K} \begin{pmatrix} \Psi_3^v & \Psi_3^e \end{pmatrix} \right), \tag{22}$$

$$E_{\text{Init}} = \frac{F^2}{K} \begin{pmatrix} \Psi_2^v & \Psi_2^e \end{pmatrix}, \tag{23}$$

$$E_{\text{Samp}} = F^2 \begin{pmatrix} \Psi_2^e & \Psi_2^d \end{pmatrix}, \tag{24}$$

$$E_{\text{Bias}} = F^2 \begin{pmatrix} 1 & 2\Psi_1 + \Psi_2^d \end{pmatrix}. \tag{25}$$

These equations show that ensembling only affects the noise and initialization variances. In both cases, their divergence at the interpolation threshold (due to Ψ_2^v, Ψ_3^v) is suppressed as $1/K$, see figure 6 for an illustration. At $P > N$, ensembling and overparametrizing have a very similar effect: they wipe out these two troubling sources of randomness by averaging them out over more random features. Indeed, we see in figure 5 that in this overparametrized regime, sending $K \to \infty$ has the same effect as sending $P/N \to \infty$: in both cases the system approaches the kernel limit. At $P < N$, this is not true: as shown in (Jacot et al., 2020), the $K \to \infty$ predictor still operates in the kernel limit, but with an effective regularization parameter $\tilde{\lambda} > \lambda$ which diverges as $P/N \to 0$. This (detrimental) implicit regularization increases the test error.

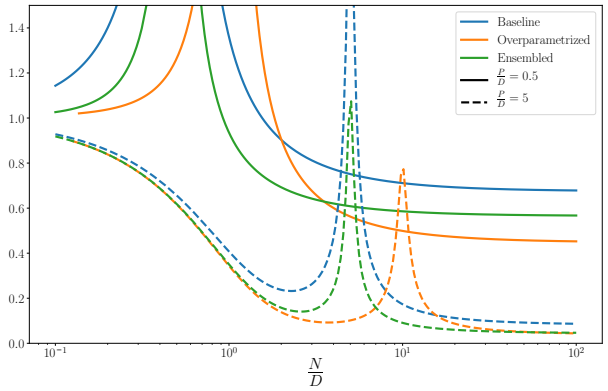

Figure 7. Comparison of the test error of a RF model (blue) with that obtained by doubling the number features (orange) or ensembling over two initializations of the features (green), as function of N/D. The parameters are $\lambda = 10^{-5}$, SNR $= 10$, $P/D = 0.5$ (solid lines) and $P/D = 5$ (dashed lines).

Ensembling vs. overparametrization As we have shown, ensembling and overparametrizing have similar effects in the lazy regime. But which is more powerful: ensembling K models, or using a single model with K times more features? The answer is given in figure 7 for $K = 2$ where we plot our analytical results while varying the number of data points, N. Two observations are particularly interesting. First, overparametrization shifts the interpolation threshold, opening up a region where ensembling outperforms overparametrizing. Second, overparametrization yields a higher asymptotic improvement in the large dataset limit $N/D \to \infty$, but the gap between overparametrizing and ensembling is reduced as P/D increases. At $P \gg D$, where we are already close to the kernel limit, both methods yield a similar improvement. Note that from the point of view of efficiency, ridge regression involves the inversion of a $P \times P$ matrix, therefore ensembling is significantly more efficient.

Ensembling vs. optimal regularization In all the results presented above, we keep the regularization constant λ fixed. However, by appropriately choosing the value of λ at each value of P/N, the performance is improved. As figure 7 reveals, the optimal value of λ decreases with K since the minimum of the test error shifts to the left when increasing K. In other words, ensembling is best when the predictors one ensembles upon are individually under-regularized, as was observed previously for kernel learning in (Zhang et al., 2015).

Through a comparison between the performance, in test error perspective, of the ensembled system with $K \to \infty$ and of a single RF model ($K = 1$) optimally regularized, figure 8 shows that the ensembled system always performs better than the optimally regularized one.

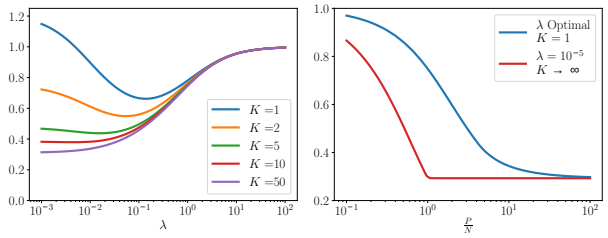

Figure 8. **Left:** Test error as a function of λ for various values of K and parameters $P/D = 2$, $N/D = 1$, SNR $= 10$. **Right:** Comparison of test error for an optimal regularized system with $K = 1$ and the system with $K \to \infty$ with $\lambda = 10^{-5}$. Optimization performed over 50 values of λ from 10^{-5} to 10^{2}. Parameters are $N/D = 1$, SNR $= 10$.

6. Numerical experiments on neural networks

Finally, we investigate whether the phenomenology described here holds for realistic neural networks learning real data in the lazy regime.

We train a 5-layer fully-connected network on the CIFAR-10 dataset. We keep only the first ten PCA components of the images, and divide the images in two classes according to the parity of the labels. We perform 10^5 steps of full-batch gradient descent with the *Adam* optimizer and a learning rate of 0.1, and scale the weights as prescribed in (Jacot et al., 2018).

We gradually go from the usual *feature learning* regime to the *lazy learning* regime using the trick introduced in (Chizat et al., 2019), which consists in scaling the output of the network by a factor α and replacing the learning function $f_\theta(\boldsymbol{x})$ by $\alpha(f_\theta(\boldsymbol{x}) - f_{\theta_0}(\boldsymbol{x}))$. For $\alpha \gg 1$, one must have that $\theta - \theta_0 \sim 1/\alpha$ in order for the learning function to remain of order one. In other words, the weights are forced to stay close to their initialization, hence the name *lazy learning*.

Results are shown in figure 9. Close to the lazy regime ($\alpha = 100$, right panel), a very similar behavior as the RF model is observed. The test error curve[7] obtained when ensembling $K = 20$ independently initialized networks becomes roughly flat after the interpolation threshold (which here is signalled by the peak in the test accuracy). As we move away from the lazy regime ($\alpha = 10$, left panel), the same curve develops a *dip* around the interpolation threshold and increases beyond $P > N$ as observed previously in (Geiger et al., 2019a). This may arguably be associated to the beneficial effect of *feature learning*.

Acknowledgements We thank Matthieu Wyart and Lenka Zdeborová for discussions related to this project. This work is supported by the French Agence Nationale de la

[7] Note that we are considering a binary classification task here: the error is defined as the fraction of misclassified images.

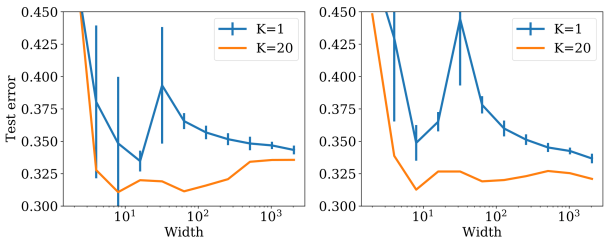

Figure 9. Test error on the binary 10-PCA CIFAR10 as function of the number of nodes per layer of the 5-layer neural network trained until convergence with the full-batch *Adam*. We compare the test error of a single predictor ($K = 1$), averaged over 20 initializations of the weights (the standard deviation is depicted as vertical bars), with the ensembling predictor at $K = 20$. **Left:** $\alpha = 10$. **Right:** $\alpha = 100$, where we are closer to the lazy regime and the ensembling curve flattens beyond the interpolation threshold, which occurs around 30 nodes per layer.

Recherche under grant ANR-17-CE23-0023-01 PAIL and ANR-19-P3IA-0001 PRAIRIE, and by the Simons Foundation (#454935, Giulio Biroli). We also acknowledge support from the chaire CFM-ENS "Science des données".

Upon completion of this paper, we became aware of two related parallel works presenting bias-variance tradeoffs for RF models. The variance due to the sampling of the dataset was considered in (Yang et al., 2020), whereas (Jacot et al., 2020) focused on the variance due to the randomness of the random feature vectors.

References

Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. *Learning from data*, volume 4. AMLBook New York, NY, USA:, 2012.

Advani, M., Lahiri, S., and Ganguli, S. Statistical mechanics of complex neural systems and high dimensional data. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(03):P03014, 2013.

Advani, M. S. and Saxe, A. M. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

Aggarwal, A., Lopatto, P., and Yau, H.-T. Goe statistics for levy matrices. *arXiv preprint arXiv:1806.07363*, 2018.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019a.

Arora, S., Du, S. S., Li, Z., Salakhutdinov, R., Wang, R., and Yu, D. Harnessing the power of infinitely wide deep nets on small-data tasks. *arXiv preprint arXiv:1910.01663*, 2019b.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine learning and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118*, 2018.

Belkin, M., Hsu, D., and Xu, J. Two models of double descent for weak features. *arXiv preprint arXiv:1903.07571*, 2019.

Breiman, L. Reflections after refereeing papers for nips. *The Mathematics of Generalization*, pp. 11–15, 1995.

Bun, J., Bouchaud, J.-P., and Potters, M. Cleaning correlation matrices. *Risk magazine*, 2015, 2016.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 2933–2943. Curran Associates, Inc., 2019.

Daniely, A., Frostig, R., and Singer, Y. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, pp. 2253–2261, 2016.

d'Ascoli, S., Sagun, L., and Biroli, G. Triple descent and the two kinds of overfitting: Where & why do they appear? *arXiv preprint arXiv:2006.03509*, 2020.

Deng, Z., Kammoun, A., and Thrampoulidis, C. A model of double descent for high-dimensional binary linear classification. *arXiv preprint arXiv:1911.05822*, 2019.

Drucker, H., Cortes, C., Jackel, L. D., LeCun, Y., and Vapnik, V. Boosting and other ensemble methods. *Neural Computation*, 6(6):1289–1301, 1994.

Engel, A. and Van den Broeck, C. *Statistical mechanics of learning*. Cambridge University Press, 2001.

Franz, S. and Parisi, G. The simplest model of jamming. *Journal of Physics A: Mathematical and Theoretical*, 49 (14):145001, 2016.

Geiger, M., Jacot, A., Spigler, S., Gabriel, F., Sagun, L., d'Ascoli, S., Biroli, G., Hongler, C., and Wyart, M. Scaling description of generalization with number of parameters in deep learning. *arXiv preprint arXiv:1901.01608*, 2019a.

Geiger, M., Spigler, S., Jacot, A., and Wyart, M. Disentangling feature and lazy learning in deep neural networks: an empirical study. *arXiv preprint arXiv:1906.08034*, 2019b.

Gerace, F., Loureiro, B., Krzakala, F., Mézard, M., and Zdeborová, L. Generalisation error in learning with random features and the hidden manifold model. *arXiv preprint arXiv:2002.09339*, 2020.

Goldt, S., Mézard, M., Krzakala, F., and Zdeborová, L. Modelling the influence of data structure on learning in neural networks. *arXiv preprint arXiv:1909.11500*, 2019.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Jacot, A., Şimşek, B., Spadaro, F., Hongler, C., and Gabriel, F. Implicit regularization of random feature models. *arXiv preprint arXiv:2002.08404*, 2020.

Kini, G. and Thrampoulidis, C. Analytic study of double descent in binary classification: The impact of loss. *arXiv preprint arXiv:2001.11572*, 2020.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Krzakala, F. and Kurchan, J. Landscape analysis of constraint satisfaction problems. *Physical Review E*, 76(2): 021122, 2007.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Livan, G., Novaes, M., and Vivo, P. *Introduction to random matrices: theory and practice*, volume 26. Springer, 2018.

Mei, S. and Montanari, A. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

Mei, S., Misiakiewicz, T., and Montanari, A. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. *arXiv preprint arXiv:1902.06015*, 2019.

Mézard, M., Parisi, G., and Virasoro, M. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.

Nakkiran, P. More data can hurt for linear regression: Sample-wise double descent. *arXiv preprint arXiv:1912.07242*, 2019.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

Neal, B., Mittal, S., Baratin, A., Tantia, V., Scicluna, M., Lacoste-Julien, S., and Mitliagkas, I. A modern take on the bias-variance tradeoff in neural networks. *arXiv preprint arXiv:1810.08591*, 2018.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Neyshabur, B., Tomioka, R., Salakhutdinov, R., and Srebro, N. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*, 2017.

Opper, M. and Kinzel, W. Statistical mechanics of generalization. In *Models of neural networks III*, pp. 151–209. Springer, 1996.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Rosset, S., Zhu, J., and Hastie, T. J. Margin maximizing loss functions. In *Advances in neural information processing systems*, pp. 1237–1244, 2004.

Seung, H. S., Sompolinsky, H., and Tishby, N. Statistical mechanics of learning from examples. *Physical review A*, 45(8):6056, 1992.

Spigler, S., Geiger, M., d'Ascoli, S., Sagun, L., Biroli, G., and Wyart, M. A jamming transition from under-to over-parametrization affects loss landscape and generalization. *arXiv preprint arXiv:1810.09665*, 2018.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Tarquini, E., Biroli, G., and Tarzia, M. Level statistics and localization transitions of levy matrices. *Physical review letters*, 116(1):010601, 2016.

Woodworth, B., Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Kernel and deep regimes in overparametrized models. *arXiv preprint arXiv:1906.05827*, 2019.

Yang, Z., Yu, Y., You, C., Steinhardt, J., and Ma, Y. Rethinking bias-variance trade-off for generalization of neural networks. *arXiv preprint arXiv:2002.11328*, 2020.

Zdeborová, L. and Krzakala, F. Phase transitions in the coloring of random graphs. *Physical Review E*, 76(3): 031131, 2007.

Zdeborová, L. and Krzakala, F. Statistical physics of inference: Thresholds and algorithms. *Advances in Physics*, 65(5):453–552, 2016.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Zhang, Y., Duchi, J., and Wainwright, M. Divide and conquer kernel ridge regression. In *Conference on Learning Theory*, pp. 592–617, 2013.

Zhang, Y., Duchi, J., and Wainwright, M. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.