High-dimensional Optimization

Courtney Paquette[*†] Elliot Paquette[†]

1 Introduction

Optimization problem structure plays an important role in designing efficient algorithms. A common structure, motivated by empirical risk minimization (ERM), is the finite sum, that is, an optimization problem of the form

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \left\{ f(\boldsymbol{x}) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{x}) \right\}, \tag{1}$$

where the functions $f_i : \mathbb{R}^d \to \mathbb{R}$. Much has been written about the complexity of stochastic and deterministic algorithms for solving (1) under various general assumptions on f, such as smoothness and convexity [7, 10, 11, 15, 19–21, 25, 26, 29–31, 39, 41, 48, 49].

Motivated in part by the rise in machine learning, optimization research has focused on weakening these assumptions, as the problems of interest are both nonconvex and nonsmooth. This has led to tight upper bounds on complexity that match information theoretic lower bounds for very general finite-sum problems [4, 23, 24, 42], and yet in spite of this, there exists an enormous gap between these theoretical guarantees and observed performance in machine learning.

Indeed, even in the smooth, convex setting, there is a missing component in our understanding of finite sum problems in machine learning. One possibility is simply the size of the finite sums. An overarching trend in machine learning is to scale problems up, engineering solutions for the next order of magnitude in problem size, measured both by model complexity and data set size. In short, machine learning problems are *high-dimensional*.

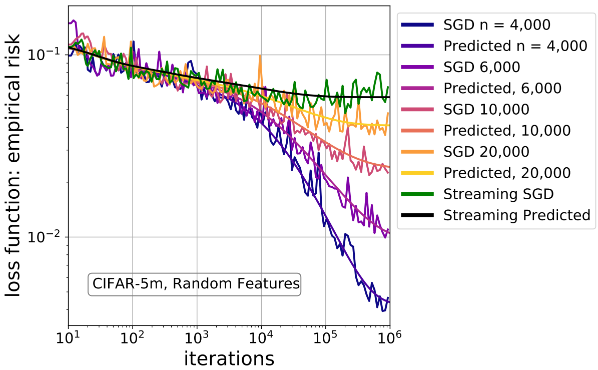

Figure 1: **Single runs of SGD vs. predicted dynamics (solid line)** on standarized CIFAR-5M [40] with car/plane class vector (1, 000, 000 samples); a standarized ReLU, random features model was applied with increasing number of samples n and fixed $d = 6000$. The predicted behavior, denoted by "predicted" (solid lines), without running SGD, matches the performance of single runs of SGD for finite n and streaming ($n = \infty$). See details in [46].

Another aspect of machine learning problems, besides that they are high dimensional, is that they are all stochastic: the data are random, the learning algorithms are random, and the model initialization is random. We propose that this trifecta of randomness combined with high-dimensionality are the missing structure from a theory of optimization for machine learning.

The purpose of this article is to survey the recent advances in developing a framework that incorporates high-dimensionality for analyzing stochastic learning algorithms on a ℓ^2-regularized least squares problem [34, 45, 46]. The main idea, discussed in detail in Section 2, is to import mathematical ideas commonly used in random matrix theory. The resulting framework yields predictions for learning curves that are amenable to analysis and often ex-

[*] Google Research, Brain Team

[†] Department of Mathematics and Statistics, McGill University, Montreal, QC; CP is a CIFAR AI chair, MILA and CP was supported by a Discovery Grant from the Natural Science and Engineering Research Council (NSERC) of Canada and a Fonds de recherche du Québec (FRQNT) Research Support for New Academics Grant; website https://cypaquette.github.io/ and email courtney.paquette@mcgill.ca. Research by EP was supported by a Discovery Grant from the Natural Science and Engineering Research Council (NSERC) of Canada; website https://elliotpaquette.github.io/ and email: elliot.paquette@mcgill.ca.

actly reproduce the behavior seen by popular algorithms (*e.g.*, stochastic gradient descent (SGD) [48]) on real data sets (see *e.g.*, Figure 1 and Figure 4). Finally we illustrate how one can use these predictions to draw important insights on average-case complexity and parameter selections such as learning rate, momentum parameter, and batch size (Section 4).

The use of tools from high-dimensional probability and random matrix theory for simplifying the analysis of optimization algorithms is a relatively new area in the machine learning literature [9, 14, 17, 18, 38, 50]. It has been used to model phenomena that, up until this point, had only been observed in deep neural networks (e.g., double descent), but which through random matrix theory are revealed to be an artifact of high-dimensional data [1, 2, 8, 22, 28, 35, 37, 52]. Beyond this, statistical assumptions to reduce complexity of analyzing algorithms were notably used in the compressed sensing community (see, for example, [12, 43]).

2 Problem Set-Up

In this section, we develop ideas from random matrix theory for incorporating high-dimensionality into the analysis of learning algorithms. To formalize the analysis, we define the *ℓ^2-regularized least squares problem*:

$$\operatorname*{arg\,min}_{\boldsymbol{x}\in\mathbb{R}^d}\Big\{ f(\boldsymbol{x}) \overset{\text{def}}{=} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|^2 + \frac{\delta}{2}\|\boldsymbol{x}\|^2 = \sum_{i=1}^n \underbrace{\frac{1}{2}\left((\boldsymbol{a}_i\boldsymbol{x}-b_i)^2 + \frac{\delta}{n}\|\boldsymbol{x}\|^2\right)}_{\overset{\text{def}}{=} f_i(\boldsymbol{x})} \Big\}. \tag{2}$$

The design matrix $\boldsymbol{A}$ is size $n \times d$, both of which we take to be large, matching the idea that both the data size (n) and the model complexity d are big. The fixed parameter $\delta > 0$ controls the regularization strength and it is independent of n and d. We do not require that n and d are proportional. Instead, we need the following:

Assumption 2.1 (Polynomially related)**.** *There is an $\alpha \in (0,1)$ so that*

$$d^\alpha \le n \le d^{1/\alpha}.$$

The data matrix $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ and the labels $\boldsymbol{b}$ may be deterministic or random; we formulate the theorems for deterministic $\boldsymbol{A}$ and $\boldsymbol{b}$ in (2) satisfying various assumptions, and in the applications of these theorems to statistical settings, we shall give examples of random $\boldsymbol{A}$ and $\boldsymbol{b}$ which satisfy these assumptions. These assumptions are motivated by the empirical risk minimization problem (ERM), and in particular the case where the augmented matrix $[\boldsymbol{A} \mid \boldsymbol{b}]$ has rows that are independent and sampled from some common distribution (see Section 2.1 for details). We also note that the problem (2) is homogeneous, in that if we simultaneously divide $\boldsymbol{A}$, $\boldsymbol{b}$ and $\sqrt{\delta}$ by any desired scalar, we produce an equivalent optimization problem. As such, we may also adopt the following normalization convention without loss of generality.

Assumption 2.2 (Data–target normalization)**.** *There is a constant $C > 0$ independent of d and n such that the spectral norm of $\boldsymbol{A}$ is bounded by C and the target vector $\boldsymbol{b} \in \mathbb{R}^n$ is normalized so that $\|\boldsymbol{b}\|^2 \le C$.*

2.1 Detour into random matrix theory

We are looking for deterministic assumptions on $(\boldsymbol{A}, \boldsymbol{b})$ that capture the combination of the high-dimensionality of the problem with the intrinsic randomness seen in a machine learning setup.

As a motivating test case, consider the Gaussian design case $\boldsymbol{A} = \boldsymbol{Z}\sqrt{\boldsymbol{\Sigma}}$ for a covariance matrix $\boldsymbol{\Sigma}$. For the target, consider the generative model with noise $\boldsymbol{b} = \boldsymbol{A}\tilde{\boldsymbol{x}}_0 + \boldsymbol{\eta}$ for $\tilde{\boldsymbol{x}}_0$ and $\boldsymbol{\eta}$ independent of $\boldsymbol{A}$.

The Gaussian matrix $\boldsymbol{A}$ enjoys some spectacular distributional invariances. Most relevant here, it satisfies that for any $n \times n$ orthogonal matrix $\boldsymbol{O}$ the distribution of $\boldsymbol{O}\boldsymbol{A}$ is the same as the distribution of $\boldsymbol{A}$. It follows as a consequence that in a singular value decomposition of $\boldsymbol{A} = \boldsymbol{U}\sqrt{\boldsymbol{\Lambda}}\boldsymbol{V}^T$, the matrix $\boldsymbol{U}$ is independent of $\boldsymbol{\Lambda}$ and $\boldsymbol{V}$ and moreover it can be taken uniformly distributed on the orthogonal group. For non-identity covariance $\boldsymbol{\Sigma}$, the same is not generally true of the matrix of right–singular vectors $\boldsymbol{V}$.

This means that the left singular vectors of $\boldsymbol{A}$ reveal nothing on either $\tilde{\boldsymbol{x}}$ or $\boldsymbol{\Sigma}$. It is, however, a type of optimization *structure*; and we shall further illustrate that this uniform distribution has profound consequences for the behavior of algorithms which operate on batch subproblems (in particular mini-batch SGD).

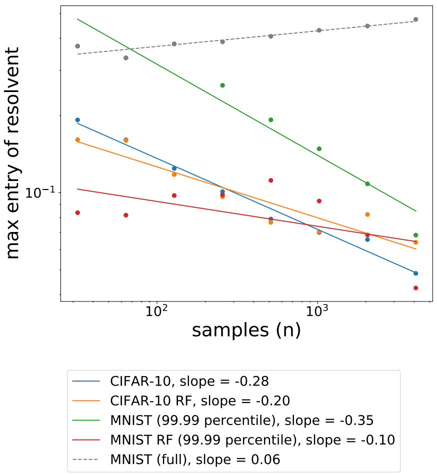

Figure 2: **Maximum off-diagonal entry of the resolvent for CIFAR-5M [40] and MNIST [33] data sets** with features d fixed (3072 and 784, respectively), varying samples $n = 2^k$ for $k = 5, 6, \ldots, 12$. Random features (RF) model was employed with $n_0 = 2000$. In the MNIST (99.99 percentile) data set large resolvent outliers were removed; when outliers not removed, MNIST data set does not satisfy the off-diagonal resolvent condition (5). For the other data sets, the off-diagonal resolvent condition is satisfied. The theory still works well for MNIST without modification (see Figure 4), which suggests that (5) could be weakened.

On the other hand, this is far too much to ask for a general design $\boldsymbol{A}$, even for one with strong statistical assumptions such as independent identically distributed subgaussian rows. We would like to generalize this assumption and ideally identify a deterministic condition which captures some of the consequences of this uniform distribution of eigenvectors.

One of the key tools in random matrix theory, especially in the theory of *universality*[1], is the *resolvent* of a matrix $\boldsymbol{M}$ defined by

$$R(z; \boldsymbol{M}) = (z\mathbf{I} - \boldsymbol{M})^{-1} \quad z \in \mathbb{C} \setminus \sigma(\boldsymbol{M}), \tag{3}$$

with $\sigma(\boldsymbol{M})$ its spectrum.

Given a singular value decomposition for $\boldsymbol{A} = \boldsymbol{U}\sqrt{\boldsymbol{\Lambda}}\boldsymbol{V}^T$, this can be computed in terms of the singular vectors by

$$\begin{aligned} &R(z; \boldsymbol{A}\boldsymbol{A}^T) = \boldsymbol{U}(z\mathbf{I} - \boldsymbol{\Lambda})^{-1}\boldsymbol{U}^T \\ \text{and} \quad &R(z; \boldsymbol{A}^T\boldsymbol{A}) = \boldsymbol{V}(z\mathbf{I} - \boldsymbol{\Lambda})^{-1}\boldsymbol{V}^T. \end{aligned} \tag{4}$$

Hence in the case of the Gaussian design, the resolvent $R(z; \boldsymbol{A}\boldsymbol{A}^T)$ factorizes as a conjugation. Moreover $\boldsymbol{U}$ is independent of $\boldsymbol{\Lambda}$ and $\boldsymbol{U}$ is uniformly distributed over the orthogonal group.

Thus the off-diagonal and diagonal entries of $R(z; \boldsymbol{A}\boldsymbol{A}^T)$ can be estimated by

$$\begin{aligned} &R(z; \boldsymbol{A}\boldsymbol{A}^T)_{ii} \approx \frac{1}{n} \operatorname{tr} R(z; \boldsymbol{\Lambda}) \\ \text{and} \quad &|R(z; \boldsymbol{A}\boldsymbol{A}^T)_{ij}| \lesssim n^{-1/2}. \end{aligned} \tag{5}$$

Further, these estimates hold with very high probability and uniformly over all i, j. The same does not hold for the other resolvent $R(z; \boldsymbol{A}^T\boldsymbol{A})$ on account of the non-uniform distribution of its singular vectors (save for in the special case that $\boldsymbol{\Sigma}$ is scalar). See Figure 2 for the behavior of the off-diagonal entry (5) on some popular data sets.

The property (5) generalizes to many other classes of random matrices with independent rows. This leads us to pose a family of assumptions on $\boldsymbol{A}$ and $\boldsymbol{b}$ which encode *some* of the flavor of the uniform distribution of left singular vectors of $\boldsymbol{A}$.

Assumption 2.3. *Suppose* Ω *is the contour enclosing* $[0, 1 + \|\boldsymbol{A}\|^2]$ *at distance* $1/2$. *Suppose there is a* $\theta \in (0, \frac{1}{2})$ *for which*

1. $\max_{z \in \Omega} \max_{1 \le i \le n} |\boldsymbol{e}_i^T R(z; \boldsymbol{A}\boldsymbol{A}^T)\boldsymbol{b}| \le n^{\theta - 1/2}$.
2. $\max_{z \in \Omega} \max_{1 \le i \ne j \le n} |\boldsymbol{e}_i^T R(z; \boldsymbol{A}\boldsymbol{A}^T)\boldsymbol{e}_j^T| \le n^{\theta - 1/2}$.
3. $\max_{z \in \Omega} \max_{1 \le i \le n} |\boldsymbol{e}_i^T R(z; \boldsymbol{A}\boldsymbol{A}^T)\boldsymbol{e}_i - \frac{1}{n} \operatorname{tr} R(z; \boldsymbol{A}\boldsymbol{A}^T)| \le n^{\theta - 1/2}$.

The second two assumptions thus encode, in a weakened form, a consequence of the uniform distribution of left-singular vectors. The first assumption additionally adds the interaction of the data matrix with the target vector $\boldsymbol{b}$. In the Gaussian

[1] Universality is the property of random matrices by which eigenvalue and eigenvector statistics are common across all large matrices having a given first and second moment structure (and sometimes 3rd and 4th) in their entries. For example, sample covariance matrices, in which $\boldsymbol{A} = \boldsymbol{W}\sqrt{\boldsymbol{\Sigma}}$ for a matrix of iid mean 0 variance 1 entries, are known to have many common spectral properties.

design together with the generative model example given above, it is also easily checked that this holds. This assumption effectively shows that individual samples have a controlled influence on solving the minimization problem, which in some sense quantifies that we are dealing with a high-dimensional problem.

The relevance of the contour enclosing the spectrum is that this allows us to pass, by contour integration, from resolvent estimates to estimates of other matrix functions, such as the ones that appear in describing the trajectories of first order algorithms. For random matrices, there is rarely a special contour of importance, and moreover the complexity of checking that the bound on a contour is roughly the same as checking the bound holds at any z with a minimum separation from the spectrum of $\boldsymbol{A}\boldsymbol{A}^T$.

2.2 Algorithmic setup

Stochastic learning algorithms and their momentum variants are the work horses in machine learning due to their relatively cheap computational cost and simple implementations.

We solve the ℓ^2-regularized least-squares problem (2) using stochastic learning algorithms, and in particular, stochastic gradient descent (SGD) with learning rate γ_k. For an initial vector $\boldsymbol{x}_0 \in \mathbb{R}^d$, we define a sequence of SGD iterates $\{\boldsymbol{x}_k\}_{k=0}^{\infty}$ which obey the recurrence,

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \gamma_k \nabla f_{i_k}(\boldsymbol{x}_k) \\ &= \boldsymbol{x}_k - \gamma_k \boldsymbol{A}^T \boldsymbol{e}_{i_k} \boldsymbol{e}_{i_k}^T (\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}) - \tfrac{\gamma_k \delta}{n} \boldsymbol{x}_k \,. \end{aligned} \tag{6}$$

The rows $\{i_1, i_2, \dots\}$ are chosen uniformly at random, and thus the batch size is one. The work of [44] suggests that under similar (albeit more restrictive) assumptions, minibatch SGD with batch-size $\beta = o(n)$ produces the same dynamical behavior as SGD after sampling single-batch SGD at iteration counts $\beta\mathbb{N}$. Therefore, we content ourselves with the simpler case with batch size equal to one. In Section 4, we will explore the effects of large batches on SGD and its momentum variant, but for now, we consider only $\beta = o(n)$.

As we want to give descriptions of the dynamics of SGD which are consistent across increasing dimensions, we suppose that γ_k has a smoothly varying schedule. Specifically, we suppose:

Assumption 2.4. *There is a continuous bounded function $\gamma : [0,\infty) \to [0,\infty)$ such that $\gamma_k = \gamma(k/n)$ for all k. As such*

$$\widehat{\gamma} \overset{def}{=} \sup_t \gamma(t) < \infty.$$

Although the classic Robbins-Monro $\gamma_k = \frac{1}{k}$ does not technically fit into this framework, for problems in which Assumptions 2.2 and 2.3 are in effect (or more generally where some non-trivial fraction of the samples are needed to commence learning), the classic $1/k$ rate is often too slow to produce any practically relevant results. Moreover, from a theoretical point of view, such a rate produces a behavior similar to gradient flow (see (10)), and it could be viewed as effectively non-stochastic. In our high-dimensional setting, a suitable analogue of the Robbins-Monro schedule that does satisfy our assumptions and yields nontrivial behavior is $\gamma_k = \frac{n}{n+k} = \frac{1}{1+k/n}$.

As for the initialization $\boldsymbol{x}_0$, we need to suppose that it does not interact too strongly with the *right* singular-vectors of $\boldsymbol{A}$. In the spirit of Assumption 2.3, it suffices to assume the following:

Assumption 2.5. *Let Ω be the same contour as in Assumption 2.3 and let $\theta \in (0, \frac{1}{2})$. Then*

$$\max_{z \in \Omega} \max_{1 \le i \le d} |\boldsymbol{e}_i^T R(z; \boldsymbol{A}^T\boldsymbol{A}) \boldsymbol{x}_0| \le n^{\theta - 1/2}.$$

Note that, as a simple but common case, this assumption is surely satisfied for $\boldsymbol{x}_0 = \boldsymbol{0}$. In principle, this assumption is general enough to allow for $\boldsymbol{x}_0$ which are correlated with $\boldsymbol{A}$ in a nontrivial way, but we do not have an application for such an initialization. For a large class of nonzero initializations independent from $(\boldsymbol{A}, \boldsymbol{b})$, this assumption is satisfied, as a corollary of Assumption 2.3:

Lemma 2.6. *Suppose that Assumption 2.3 holds with some $\theta_0 \in (0, \frac{1}{2})$ and that $\boldsymbol{x}_0$ is chosen randomly, independent of $(\boldsymbol{A}, \boldsymbol{b})$, and with independent coordinates in such a way that for some C independent of d or n*

$$\begin{aligned} &\|\mathbb{E}\,\boldsymbol{x}_0\|_\infty \le C/n \\ \textit{and} \quad &\max_i \|(\boldsymbol{x}_0 - \mathbb{E}\,\boldsymbol{x}_0)_i\|_{\psi_2}^2 \le C n^{2\theta_0 - 1}. \end{aligned}$$

Assumption 2.5 holds with any $\theta > \theta_0$ on an event of probability tending to 1 as $n \to \infty$.

Note that this assumption allows for deterministic $\boldsymbol{x}_0$ having maximum norm $\mathcal{O}(1/n)$, as well as iid centered subgaussian vectors of Euclidean norm $\mathcal{O}(1)$.

2.3 Examples of data matrix and target

We highlight two examples for which the data matrix $\boldsymbol{A}$ and target vector $\boldsymbol{b}$ satisfy Assumption 2.3 (proofs found in [46, Lemma 1.3 and Theorem 1.7]). In both cases below, we take the initialization vector $\boldsymbol{x}_0$ to be iid centered subgaussian with $\mathbb{E}\left[\|\boldsymbol{x}_0\|^2\right] = \widehat{R}$ for $\widehat{R} > 0$. Assumption 2.4 holds for this initialization vector.

Sample covariance matrices and generative models. Suppose that $\boldsymbol{\Sigma} \succeq 0$ is a $d \times d$ matrix with $\operatorname{tr}\boldsymbol{\Sigma} = 1$ and $\|\boldsymbol{\Sigma}\| \le M/\sqrt{d} < \infty$. The data matrix $\boldsymbol{A}$ is a random matrix with $\boldsymbol{A} = \boldsymbol{Z}\sqrt{\boldsymbol{\Sigma}}$ where $\boldsymbol{Z}$ is an $n \times d$ matrix of independent, mean 0, variance 1 entries with subgaussian norm at most $M < \infty$, and we assume $n \le Md$. Finally suppose that $\boldsymbol{b}$ satisfies a generative model, that is $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{\beta} + \boldsymbol{\eta}$ for $\boldsymbol{\beta}, \boldsymbol{\eta}$ iid centered subgaussian satisfying $\|\boldsymbol{b}\|^2 = R$ and $\|\boldsymbol{\eta}\|^2 = \tilde{R}\frac{n}{d}$ for some $\tilde{R}, R > 0$. For data matrix $\boldsymbol{A}$ and target vector generated this way, Assumption 2.3 holds.

Random features model of a linear ground truth. We follow the set-up based upon [1, 37]. This model encompasses two-layer neural networks with a squared loss, where the first layer has random weights and the second layer's weights are given by the regression coefficients. Suppose the $n \times n_0$ data matrix $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{\Sigma}^{1/2}/\sqrt{n_0}$ for an iid standard Gaussian $\boldsymbol{Z}$ and the covariance matrix $\boldsymbol{\Sigma}$ satisfies $1/n_0 \operatorname{tr}(\boldsymbol{\Sigma}) = 1$ and $\|\boldsymbol{\Sigma}\| \le C$ for some $C > 0$. We also suppose that $\boldsymbol{W}$ is an $n_0 \times d$ iid feature matrix having standard Gaussian entries and independent of $\boldsymbol{Z}$ so that $\boldsymbol{X}\boldsymbol{W}$ is a matrix whose rows are standardized. We now apply an activation function entry-wise. The activation function σ satisfies for $C_0, C_1 \ge 0$

$$\begin{gathered} |\sigma'(x)| \le C_0 e^{C_1|x|}, \quad \text{for all } x \in \mathbb{R} \\ \text{and for all } Z \sim N(0,1), \quad \mathbb{E}\left[\sigma(Z)\right] = 0. \end{gathered} \tag{7}$$

We now transform the data $\boldsymbol{X} \in \mathbb{R}^{n_0 \times d}$ by putting

$$\boldsymbol{A} = \sigma(\boldsymbol{X}\boldsymbol{W}/\sqrt{n_0}) \in \mathbb{R}^{n \times d}.$$

For the target vector $\boldsymbol{b}$, we use a linear ground truth model, that is, $\boldsymbol{b} = \boldsymbol{X}\boldsymbol{\beta} + \eta\boldsymbol{w}$ with $\boldsymbol{\beta}, \boldsymbol{w}$ independent isotropic subgaussian vectors with $\mathbb{E}\left[\|\boldsymbol{\beta}\|^2\right] = 1/n_0$ and $\mathbb{E}\left[\|\boldsymbol{w}\|^2\right] = 1$ and η bounded, independent of n. Assumption 2.3 holds for $\boldsymbol{A}$ and $\boldsymbol{b}$ generated this way.

3 Predicting learning curves

A benefit of working in high-dimensional optimization is that seemingly challenging tasks such as understanding the noise produced by SGD, become much simpler due to concentration effects. In fact, the entire training path taken by SGD concentrates around a deterministic function. The function gives a simple description of the exact learning curve of SGD, depending only on the spectrum of $\boldsymbol{A}\boldsymbol{A}^T$, the target $\boldsymbol{b}$, and initial $\boldsymbol{x}_0$. In this way, one can predict the training behavior of SGD without ever running SGD. The idea hinges on exploiting the trifecta of randomness in the problem and high-dimensionality through concentration of measure. Moreover, these predictions are amenable to analysis and one can draw important insights on typical computational complexity and parameter selection policies (see Section 4).

In a high-dimensional setting, this empirical risk concentrates around a deterministic path Ψ_t. To define this path, we introduce the integrated learning rate Γ and kernel K, for any $d \times d$ matrix $\boldsymbol{P}$,

$$\begin{aligned} &\Gamma(t) = \int_0^t \gamma(s)\ \mathrm{d}s, \quad \text{and} \\ &K(t,s;\boldsymbol{P}) = \tfrac{1}{n}\gamma^2(s)\operatorname{tr}\bigg(\boldsymbol{P}(\nabla^2\mathcal{L}) \\ &\qquad \times \exp\big(-2(\nabla^2\mathcal{L} + \delta\mathbf{I}_d)(\Gamma(t) - \Gamma(s))\big)\bigg). \end{aligned} \tag{8}$$

The path Ψ_t satisfies the Volterra integral equation:

$$\Psi_t = \mathcal{L}\big(\mathfrak{X}^{\text{gf}}_{\Gamma(t)}\big) + \int_0^t K(t,s;\nabla^2\mathcal{L})\Psi_s\,\mathrm{d}s. \tag{9}$$

The quantity $\mathfrak{X}^{\text{gf}}_t$ is *gradient flow* which is the solution to the differential equation

$$\mathrm{d}\mathfrak{X}^{\text{gf}}_t = -\nabla f(\mathfrak{X}^{\text{gf}}_t)\ \mathrm{d}t, \quad \mathfrak{X}^{\text{gf}}_0 = \boldsymbol{x}_0. \tag{10}$$

For the ℓ^2−regularized least squares problem, the solution to gradient flow is explicitly solvable in terms of $\boldsymbol{x}_0$, target $\boldsymbol{b}$, and eigenvalues of $\nabla^2\mathcal{L}$. Consequently, due to (9) and as Theorem 3.4 will show,

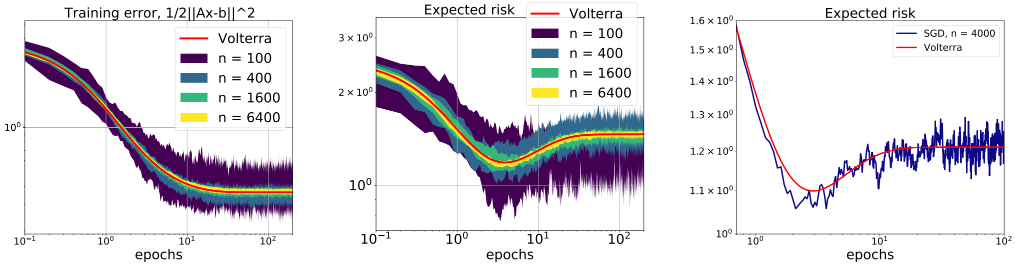

Figure 3: **Concentration of SGD on training loss and expected risk**, on a Gaussian random ℓ^2-regularized least-squares problem where $\boldsymbol{\beta} \sim N(\mathbf{0}, \mathbf{I}_d)$ is the ground truth signal and a generative model $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{\beta} + \boldsymbol{\eta}$ where entries of $\boldsymbol{\eta}$ iid standard normal with $\|\boldsymbol{\eta}\|_2^2 = 2.25$, $n = 0.9d$ with ℓ^2-regularization parameter $\delta = 0.1$, SGD was initialized at $\boldsymbol{x}_0 \sim N(\mathbf{0}, 4\mathbf{I}_d)$ (independent of $\boldsymbol{A}$, $\boldsymbol{\beta}$); an 80% confidence interval (shaded region) over 10 runs for each n, a constant learning rate for SGD was applied, $\gamma = 0.8$. For expected risk, the samples $\boldsymbol{a}$ generated from same covariance as $\boldsymbol{A}$. More volatility in the expected risk across runs even for large n in comparison to the training error (left and center). The predicted Ω_t matches the performance of SGD on the expected risk even for a single run (right).

the training dynamics of SGD are completely predictable solely from the spectrum of $\boldsymbol{A}^T\boldsymbol{A}$, target $\boldsymbol{b}$, and the initialization $\boldsymbol{x}_0$.

But one can do more. Generally, one wants to study not only the training dynamics but also the generalization performance of SGD, that is how well the algorithm performs on unseen data. In this sense, we need to be able to evaluate the iterates of SGD, trained on f (2), on other statistics. We will focus our attention on quadratics.

Definition 3.1. *A function $\mathcal{R} : \mathbb{R}^d \to \mathbb{R}$ is quadratic if it is a degree-2 polynomial or equivalently if can be represented by*

$$\mathcal{R}(\boldsymbol{x}) = \tfrac{1}{2}\boldsymbol{x}^T\boldsymbol{T}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{x} + c$$

for some $d \times d$ matrix $\boldsymbol{T}$, vector $\boldsymbol{u} \in \mathbb{R}^d$ and scalar $c \in \mathbb{R}$. For any quadratic, define the H^2–norm:

$$\begin{aligned}\|\mathcal{R}\|_{H^2} &\overset{def}{=} \|\nabla^2\mathcal{R}\| + \|\nabla\mathcal{R}(0)\| + |\mathcal{R}(0)| \\ &= \|\boldsymbol{T}\| + \|\boldsymbol{u}\| + |c|.\end{aligned} \tag{11}$$

For the learning path to concentrate on other quadratic statistics, we require an additional assumption in the same spirit as 2.3:

Assumption 3.2 (Quadratic statistics)**.** *Suppose $\mathcal{R} : \mathbb{R}^d \to \mathbb{R}$ is quadratic, i.e. there is a symmetric matrix $\boldsymbol{T} \in \mathbb{R}^{d\times d}$, a vector $\boldsymbol{u} \in \mathbb{R}^d$, and a constant $c \in \mathbb{R}$ so that*

$$\mathcal{R}(\boldsymbol{x}_t) = \tfrac{1}{2}\boldsymbol{x}_t^T\boldsymbol{T}\boldsymbol{x}_t + \boldsymbol{u}^T\boldsymbol{x}_t + c. \tag{12}$$

We assume that $\mathcal{R}$ satisfies $\|\mathcal{R}\|_{H^2} \le C$ for some C independent of n and d. Moreover, we assume the following (for the same Ω and θ) as in Assumption 2.3:

$$\max_{z,y\in\Omega}\max_{1\le i\le n} |\boldsymbol{e}_i^T\boldsymbol{A}\widehat{\boldsymbol{T}}\boldsymbol{A}^T\boldsymbol{e}_i - \tfrac{1}{n}\operatorname{tr}(\boldsymbol{A}\widehat{\boldsymbol{T}}\boldsymbol{A}^T)| \le \|\boldsymbol{T}\|n^{-\epsilon}$$

$$\text{where} \quad \begin{cases}\widehat{\boldsymbol{T}} = R(z)\boldsymbol{T}R(y) + R(y)\boldsymbol{T}R(z), \\ R(z) = R(z; \boldsymbol{A}^T\boldsymbol{A})\end{cases} \tag{13}$$

This assumption ensures that the quadratic $\mathcal{R}$ has a Hessian which is not too correlated with any of the left singular–vectors of $\boldsymbol{A}$. Establishing Assumption 3.2 can be non–trivial in the cases when the quadratic has complicated dependence on $\boldsymbol{A}$. In simple cases, (especially for the case of the empirical risk and the norm) it follows automatically from Assumption 2.3.

Lemma 3.3. *Suppose that $\mathcal{R}$ satisfies (12) with $\boldsymbol{T}$ given by a polynomial p in $\boldsymbol{A}^T\boldsymbol{A}$ (especially $\boldsymbol{I}$ and the monomial $\boldsymbol{A}^T\boldsymbol{A}$) having bounded coefficients, and suppose $\boldsymbol{u}$ and c are norm bounded independently of n or d. Then supposing Assumptions 2.2 and 2.3 for some $\theta_0 \in (0, \frac{1}{2})$, for all n sufficiently large and for any $\theta > \theta_0$, Assumption 3.2 holds.*

Thus for example $\mathcal{R} = \mathcal{L}$ will satisfy Assumption 3.2, as will the simple Euclidean vector norm $\mathcal{R} = \|\cdot\|^2$.

The trajectory $\mathcal{R}(\boldsymbol{x}_t)$ concentrates around

$$\Omega_t = \mathcal{R}(\mathfrak{X}^{\text{gf}}_{\Gamma(t)}) + \int_0^t K(t, s; \nabla^2\mathcal{R})\Psi_s \, \mathrm{d}s. \tag{14}$$

Note the trajectories of gradient flows can computed explicitly.

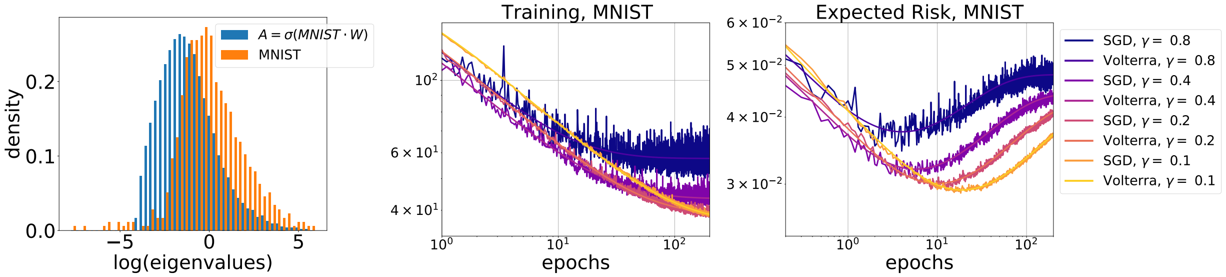

Figure 4: **SGD vs Theory on MNIST**: MNIST ($60000 \times 28 \times 28$) images. Random features model on MNIST used with $n = 4000$ images, random features $d = 2000$, and $n_0 = 28 \times 28$ was trained with one run of SGD (middle) for various learning rates and regularization parameter 0.01; entries of the random features $\boldsymbol{W}_{ij} \sim N(0,1)$ and a normalized ReLu activation function $\sigma(\cdot) = (\max\{0, \cdot\} - a)/b$ was applied. The Volterra equation matches the dynamics of the training loss (least-squares), $\mathcal{L}$, even with only one run of SGD. The log(eigenvalues) of the covariance of the MNIST dataset and the random features matrix used in the regression displayed (left). The expected risk, $\mathcal{R}(\boldsymbol{x}) = \frac{1}{2}\mathbb{E}[(b - \boldsymbol{x}^T\sigma(\boldsymbol{x}_i\boldsymbol{W}))^2]$ where $\boldsymbol{x}_i$ is an image from the MNIST test set, follows the predicted behavior Ω_t. Both the predicted Ψ_t and Ω_t match the performance of SGD in this non-idealized setting.

Finally the comparision theorem is the following:

Theorem 3.4 (Concentration of SGD)**.** *Suppose n and d are related by Assumption 2.1. Suppose the ℓ^2-regularized least-squares problem* (2) *satisfies Assumptions 2.2 and 2.3 where $n \geq d^{\tilde{\varepsilon}}$ for some $\tilde{\varepsilon} > 0$. Suppose the learning rate schedule γ satisfies Assumption 2.4, and the initialization $\boldsymbol{x}_0$ satisfies Assumption 2.5. Let $\mathcal{R} : \mathbb{R}^d \mapsto \mathbb{R}$ be any quadratic statistic satisfying Assumption 3.2. Further assume that $\mathcal{R}$ and $\mathcal{L}$ have bounded $\|\mathcal{R}\|_{H^2}$ and $\|\mathcal{L}\|_{H^2}$ independent of n or d, for some C' sufficiently large. For any deterministic $T > 0$ and any $D > 0$, there is a $C > 0$ such that*

$$\Pr\left[\sup_{0\leq t\leq T}\left\| \begin{pmatrix}\mathcal{L}(\boldsymbol{x}_{\lfloor tn\rfloor})\\ \mathcal{R}(\boldsymbol{x}_{\lfloor tn\rfloor})\end{pmatrix} - \begin{pmatrix}\Psi_t\\ \Omega_t\end{pmatrix}\right\| > d^{-\tilde{\varepsilon}/2} \,\middle|\, \boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}_0\right] \leq C'd^{-D},$$

where Ω_t solves (14) *and $\boldsymbol{x}_t$ are the iterates SGD.*

A formal proof of Theorem 3.4 can be found in [46, Theorem 1.4]. The functions Ψ_t and Ω_t can be viewed as the expected training loss and generalization error. Theorem 3.4 then shows concentration around the mean. We remark that to solve (8) we need as input $\mathcal{L}(\mathfrak{X}^{\text{gf}}_{\Gamma(t)})$ which be computed using (10).

The solution of Ψ_t can be found by repeatedly convolving the forcing term $\mathcal{L}(\mathfrak{X}^{\text{gf}}_{\Gamma(t)})$ with the kernel K (provided $\sup_{t\geq 0}\sup_{s\geq 0} K(t,s;\nabla^2\mathcal{L})$ is bounded [27]). Moreover, numerical approximations to (8) can be found by taking a large but finite number of convolutions in the expression above. The boundedness of this solution corresponds precisely to learning rate choices for which SGD is convergent.

In the case of constant learning rate $\gamma(s) \equiv \gamma$, more can be said. The Volterra equation (8) is of convolution–type, and in fact is a special case of the renewal equation [5] (allowing for *defective* and *excessive* variants). Specifically, the expression in (9) simplifies to

$$\Psi_t = \mathcal{L}(\mathfrak{X}^{\text{gf}}_{\Gamma(t)}) + \frac{\gamma^2}{n}\int_0^t \operatorname{tr}\left((\boldsymbol{A}^T\boldsymbol{A})^2 e^{-2(\boldsymbol{A}^T\boldsymbol{A}+\delta\mathbf{I}_d)(t-s)}\right)\Psi_s \,\mathrm{d}s. \tag{15}$$

In addition to fixed point algorithms, one can also use Laplace transform techniques. These solutions to (15) can be analyzed explicitly for convergence guarantees and rates of convergence, see [44, 45]. As a simple example writing $K(t,s;\boldsymbol{P}) = K(t-s;\boldsymbol{P})$, the convergence of (15) occurs precisely when $\int_0^\infty K(t;\nabla^2\mathcal{L})\,\mathrm{d}t \leq 1$.[2]

Under the assumption that $\gamma(s)$ stabilizes, i.e. $\gamma(s) \to \gamma$ as $s \to \infty$, we may still characterize the eventual behavior of the solution. In the case that $\mathcal{R}$ represents the population risk, the difference $\Omega_t - \mathcal{R}(\mathfrak{X}^{\text{gf}}_{\Gamma(t)})$ gains the interpretation of the excess risk of SGD over gradient flow. On taking

[2] See [5, Chapter V] for a general discussion. In the case that the norm is exactly 1, this remains true as it is a special case of the Blackwell renewal theorem. When the norm is larger than 1, in the event that the empirical risk of gradient flow is bounded away from 0, the training loss is divergent.

$t \to \infty$, this converges to the excess risk of the SGD estimator over the ridge regression estimator:

Theorem 3.5. *If $\gamma(t) \to 0$ but $\Gamma(t) \to \infty$ as $t \to \infty$ (c.f. the Robbins-Monro setting), then $\Omega_t - \mathcal{R}(\mathcal{X}^{gf}_{\Gamma(t)}) \xrightarrow[t\to\infty]{} 0$. If on the other hand $\gamma(t) \to \widetilde{\gamma} > 0$, where the limiting learning rate satisfies*

$$\widetilde{\gamma} < 2\big(\tfrac{1}{n}\operatorname{tr}((\boldsymbol{A}^T\boldsymbol{A})^2(\boldsymbol{A}^T\boldsymbol{A}+\delta\boldsymbol{I}_d)^{-1})\big)^{-1}, \qquad (16)$$

then with Ψ_∞ given by the limiting empirical risk:

$$\Psi_\infty = \mathcal{L}(\mathcal{X}^{gf}_\infty) \times \left(1 - \frac{\widetilde{\gamma}}{2n}\operatorname{tr}\big((\nabla^2\mathcal{L})^2(\nabla^2\mathcal{L}+\delta\boldsymbol{I}_d)^{-1}\big)\right)^{-1}$$

the limiting excess risk of SGD over ridge regression is given by

$$\Omega_t - \mathcal{R}(\mathcal{X}^{gf}_{\Gamma(t)}) \xrightarrow[t\to\infty]{} \frac{\widetilde{\gamma}}{2n}\Psi_\infty \times \operatorname{tr}\Big((\nabla^2\mathcal{R})(\nabla^2\mathcal{L})\big(\nabla^2\mathcal{L}+\delta\boldsymbol{I}_d\big)^{-1}\Big).$$

Examples of statistics. We give some common statistics that illustrate the versatility of our set-up.

One important quadratic statistic which satisfies all assumptions in Section 2.3 is $\mathcal{R}(\cdot) = \frac{1}{2}\|\cdot - \boldsymbol{\beta}\|^2$ where $\boldsymbol{\beta}$ is the unknown, ground truth signal.

Another common statistic in the standard linear regression set-up is the *population risk*. We first address the in-distribution set-up, where the data is drawn from the same distribution as the population. Let $\boldsymbol{A}$ be generated by taking n independent d-dimensional samples from a centered distribution $\mathcal{D}_f$ with feature covariance $\boldsymbol{\Sigma}_f \in \mathbb{R}^{d\times d}$, that is $\boldsymbol{\Sigma}_f = \mathbb{E}[\boldsymbol{a}\boldsymbol{a}^T]$, where $\boldsymbol{a} \sim \mathcal{D}_f$. We suppose a new data point $(\boldsymbol{a}, b)$ is drawn from a distribution $\mathcal{D}$ on $\mathbb{R}^d \times \mathbb{R}$ with the property that $\mathbb{E}[b \,|\, \boldsymbol{a}] = \boldsymbol{\beta}^T\boldsymbol{a}$ where $(\boldsymbol{a}, b) \sim \mathcal{D}$ and the data $\boldsymbol{a} \sim \mathcal{D}_f$. As before, $\boldsymbol{\beta}$ is the ground truth signal. The vector $\boldsymbol{x}_t$ generated by SGD represents an estimate of $\boldsymbol{\beta}$, and the population risk is

$$\mathcal{R}(\boldsymbol{x}_t) \overset{\text{def}}{=} \frac{1}{2}\mathbb{E}\left[(b - \boldsymbol{x}_t^T\boldsymbol{a})^2 \,|\, \boldsymbol{x}_t\right], \qquad (17)$$

where $(\boldsymbol{a}, b) \sim \mathcal{D}$.

In the case of out-of-distribution, the data matrix $\boldsymbol{A}$ is generated using one distribution but the sample $(\boldsymbol{a}, b) \sim \mathcal{D}$ is not drawn from the same distribution as $\boldsymbol{A}$, that is, $\boldsymbol{a} \sim \hat{\mathcal{D}}_f \neq \mathcal{D}_f$ but still $\mathbb{E}[b|\boldsymbol{a}] = \boldsymbol{\beta}^T\boldsymbol{a}$.

Finally for random features with a linear ground truth, we would take

$$\mathcal{R}(\boldsymbol{x}_t) \overset{\text{def}}{=} \mathbb{E}\left[(b - \boldsymbol{x}_t^T\sigma(\boldsymbol{X}_i\boldsymbol{W}/\sqrt{n_0}))^2 \,|\, \boldsymbol{x}_t, \boldsymbol{W}\right]. \qquad (18)$$

All these examples are quadratic statistics for which Theorem 3.4 applies.

4 Average-case Complexity & Parameter Selections

The dynamics of training curves for generic objective functions, in general, are quite complicated. However, as we have seen, in the case of ℓ^2-regularized least squares problem under high-dimensionality, the dynamics for stochastic learning algorithms are simple. As such, one can go further and get additional information about the performance of these algorithms. In this section, we use the predictions of the exact training dynamics to draw important insights on typical computational complexity and parameter selection (*e.g.*, learning rate, batch size, and momentum parameters). We will focus our attention on the widely used stochastic gradient descent algorithm with momentum (SGD+M) on the ℓ^2-regularized least squares problem with regularization parameter $\delta = 0$ (*e.g.*, $\min_{\boldsymbol{x}} 1/2\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\|^2$). Mini-batch SGD+M is defined by selecting uniformly at random a subset $B_k \subseteq \{1, 2, \cdots, n\}$ of cardinality β and making the update

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \gamma \sum_{i\in B_k} \nabla f_i(\boldsymbol{x}_k) + \Delta(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \\ &= \boldsymbol{x}_k - \gamma\boldsymbol{A}^T\boldsymbol{P}_k(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}) + \Delta(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}), \\ &\text{where} \quad \boldsymbol{P}_k \overset{\text{def}}{=} \sum_{i\in B_k} \boldsymbol{e}_i\boldsymbol{e}_i^T, \end{aligned} \qquad (19)$$

with $\boldsymbol{P}_k$ a random orthogonal projection matrix and $\boldsymbol{e}_i$ the i-th standard basis vector. Here $\gamma > 0$ is the learning rate parameter, Δ is the momentum parameter, and the function f_i is the i-th element of the sum in (2). Note we are only considering the constant learning rate and momentum setting. We define the *batch fraction* ζ as the ratio of β/n.

When the stochastic gradient in (19) is replaced with the full-gradient $\nabla f(\boldsymbol{x})$ and the hyperparameters are chosen optimally, the resulting algorithm is the celebrated heavy-ball momentum (a.k.a. Polyak momentum) [47]. The optimal learning rate and momentum parameters are explicitly given by

$$\gamma = \frac{4}{(\sqrt{\sigma_{\max}^2} + \sqrt{\sigma_{\min}^2})^2}$$
$$\text{and} \quad \Delta = \left(\frac{\sqrt{\sigma_{\max}^2} - \sqrt{\sigma_{\min}^2}}{\sqrt{\sigma_{\max}^2} + \sqrt{\sigma_{\min}^2}} \right)^2. \tag{20}$$

It is well-known that heavy-ball is an optimal algorithm on the least squares problem in that it converges linearly at a rate of $\mathcal{O}(1/\sqrt{\kappa})$.

In the influential work of [51], the authors empirically show that SGD+M significantly improves training performance of deep neural networks. Despite its wide usage in machine learning practice, our understanding of it is more narrow. It has been hypothesized that SGD+M improves training because it is employed on a large batch of a data set [32], thereby emulating the speed-up one sees in full-batch settings. For many learning problems, the "large batch" setting is often paired with high-dimensional problems, meaning there are many samples (and likely also many features to have interesting behavior). There have been some recent works in proving that for sufficiently large batch sizes SGD+M does achieve $\mathcal{O}(1/\sqrt{\kappa})$ [13, 16]; see also [3, 30, 36]. We can go further and find the correct batch size dependency in the high-dimensional regime.

First, we address how the batch effects the training dynamics.

Theorem 4.1 (Concentration of mini-batch momentum)**.** *Suppose the assumptions of Theorem 3.4 hold. For any deterministic $T > 0$ and any $D > 0$, there is a $C > 0$ such that*

$$\Pr\left[\sup_{0 \le k \le T} \left| \mathcal{L}(\boldsymbol{x}_k) - \Psi_k \right| > d^{-\tilde{\varepsilon}/2} \,\middle|\, \boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}_0 \right] \le C' d^{-D},$$

where Ψ_k solves a discrete convolution-type Volterra equation

$$\Psi_{k+1} = \mathcal{L}(\boldsymbol{x}_{k+1}^{\text{GD+M}(\gamma\zeta)}) + \sum_{t=0}^{k} K(k-t; \nabla^2 \mathcal{L}) \Psi_t. \tag{21}$$

Here $K(k)$ is a kernel completely determined by the spectrum of $\nabla^2 \mathcal{L}$ and $\{\boldsymbol{x}_k^{\text{GD+M}(\gamma\zeta)}\}$ are the iterates generated by running full-batched momentum (i.e., $\zeta = 1$ in (19)) with learning rate given by $\zeta\gamma$ and momentum parameter Δ.

The expression for (21) can be viewed as a discrete convolution-type Volterra equation with forcing term $\mathcal{L}(\boldsymbol{x}_k^{\text{GD+M}(\gamma\zeta)})$ and kernel $K(t; \nabla^2 \mathcal{L})$. The forcing term, $F(k) = \mathcal{L}(\boldsymbol{x}_k^{\text{GD+M}(\gamma\zeta)})$ represents the mean (with respect to expectation over the mini-batches) behavior of SGD+M. For small learning rates γ, the forcing term controls the dynamics of Ψ_k. We denote the dominant term in $F(k)$ by $\lambda_{\max}(\gamma, \Delta, \zeta)$, that is $F(k) = \mathcal{O}(\lambda_{\max}^k)$. Specifically,

$$\lambda_j \overset{\text{def}}{=} \frac{-2\Delta + \Omega_j^2 + \sqrt{\Omega_j^2(\Omega_j^2 - 4\Delta)}}{2},$$
$$\text{where} \quad \Omega_j \overset{\text{def}}{=} 1 - \gamma\zeta\sigma_j^2 + \Delta,$$
$$\lambda_{\max} \overset{\text{def}}{=} \max_{1 \le j \le n} |\lambda_j|, \text{ and } \sigma_j^2, \text{ eigenvalue of } \boldsymbol{A}\boldsymbol{A}^T. \tag{22}$$

On the other hand, the kernel term, or convolution in (21), $\sum_{t=0}^{k} K(k-t, \nabla^2 \mathcal{L}) \Psi_t$, is due to the inherent stochasity generated by uniformly at random selecting indices. The presence of Ψ_t (training loss) in this term is due to the fact that the noise generated by the k-th stochastic gradient is proportional to Ψ_t, and the function $K(k-t)$ represents the progress of the algorithm in sending this extra noise to 0. We note that the kernel $K(k-t)$ in (21) scales quadratically in the learning rate γ. Hence for *large* learning rates, the kernel dominates the decay behavior of Ψ_k.

There are also some key relationships between Theorem 3.4 (batch sizes $\zeta \to 0$) and Theorem 4.1, notably that (when $\Delta = 0$), gradient flow in the forcing term of (9) becomes gradient descent (21) – discrete gradient flow. A similar discretization is observed in the kernel with an integral replaced by a summation.

4.1 Convolution Volterra analysis

We begin by establishing sufficient conditions for the convergence of the solution to the Volterra equation (21), a special case of the *renewal equation* ([6]). Let us translate (21) into the form of the renewal equation as follows:

$$\psi(t+1) = F(t+1) + (\tilde{K} * \psi)(t), \tag{23}$$

where $(f * g)(t) = \sum_{k=0}^{\infty} f(t-k) g(k)$. Let the *kernel norm* be $\|\tilde{K}\| = \sum_{t=0}^{\infty} \tilde{K}(t)$. By [6, Proposition 7.4], we see that $\|\tilde{K}\| < 1$ is necessary for our solution to the Volterra equation to be convergent. Indeed, we have the following result.

Proposition 4.2. *If the norm $\|\tilde{K}\| < 1$, the algorithm is convergent in that*

$$\Psi_\infty \overset{def}{=} \lim_{k\to\infty} \Psi_k = \frac{\lim_{k\to\infty} \mathcal{L}(\boldsymbol{x}_k^{\text{GD+M}(\gamma\zeta)})}{1 - \|\tilde{K}\|}. \tag{24}$$

Proposition 4.2 formulates the limit behaviour of the objective function in both the over-determined and the under-determined cases of least squares. When under-determined, the limiting loss value of $\mathcal{L}(\boldsymbol{x}_k^{\text{GD+M}(\gamma\zeta)}) = 0$ and the limiting Ψ_∞ is 0; otherwise the limiting loss value is strictly positive. The result (24) only makes sense when the noise term K satisfies $||K|| < 1$; the next proposition illustrates the conditions on the learning rate and the trace of the eigenvalues of $\boldsymbol{A}\boldsymbol{A}^T$ such that the kernel norm is less than 1.

Proposition 4.3 (Convergence threshold)**.** *Under the learning rate condition $\gamma < \frac{1+\Delta}{\zeta\sigma_{max}^2}$ and trace condition $\frac{(1-\zeta)\gamma}{1-\Delta} \cdot \frac{1}{n}\operatorname{tr}(\boldsymbol{A}\boldsymbol{A}^T) < 1$, the kernel norm $\|\tilde{K}\| < 1$, i.e., $\sum_{t=0}^\infty \tilde{K}(t) < 1$.*

The *learning rate condition* quantifies an upper bound of good learning rates by the largest eigenvalue of the covariance matrix $\sigma^2_{\max}$, batch fraction ζ, and the momentum parameter Δ. The *trace condition* illustrates a constraint on the growth of $\sigma^2_{\max}$. Moreover, for a full batch gradient descent model ($\zeta = 1$), the trace condition can be dropped and we get the classical learning rate condition for gradient descent.

4.2 The Malthusian exponent and complexity

The rate of convergence of Ψ_k is essentially the worse of two terms – the forcing term $F(t)$ and a discrete time convolution $\sum_{t=0}^k K(k-t; \nabla^2\mathcal{L})\Psi_t$ which depends on the kernel K. Intuitively, the forcing term captures the behavior of the expected value of SGD+M and the discrete time convolution captures the slowdown in training due to noise created by the algorithm. Note that $F(k)$ is always a lower bound for Ψ_k, but it can be that Ψ_k is exponentially (in k) larger than $F(k)$ owing to the convolution term. This occurs when something called the *Malthusian exponent*, denoted Ξ, of the convolution Volterra equation exists. The Malthusian exponent Ξ is given as the unique solution to

$$\gamma^2\zeta(1-\zeta)\sum_{t=0}^\infty \Xi^t K(t; \nabla^2\mathcal{L}) = 1, \text{ if solution exists.} \tag{25}$$

The Malthusian exponent enters into the complexity analysis in the following way:

Theorem 4.4 (Asymptotic rates)**.** *The inverse of the Malthusian exponent always satisfies $\Xi^{-1} > \Lambda$ for finite n. Moreover, for some $C > 0$, the convergence rate for SGD+M is*

$$\begin{aligned} &\Psi_k - \Psi_\infty \le C\max\{\lambda_{\max}, \Xi^{-1}\}^k \\ \textit{and} \quad &\lim_{t\to\infty}(\Psi_k - \Psi_\infty)^{1/k} = \max\{\lambda_{\max}, \Xi^{-1}\}. \end{aligned} \tag{26}$$

Thus to understand the rates of convergence, it is necessary to understand the Malthusian exponent as a function of γ and Δ.

4.3 Two regimes for the Malthusian exponent

On the one hand, the Malthusian exponent Ξ comes from the stochasticity of the algorithm itself. On the other hand, $\lambda_{\max}(\gamma, \Delta, \zeta)$ is determined completely by the problem instance information — the eigenspectrum of $\boldsymbol{A}\boldsymbol{A}^T$. (Note we want to emphasize the dependence of $\lambda_{\max}$ on the learning rate, the momentum parameter, and the batch fraction). Let $\sigma^2_{\max}$ and $\sigma^2_{\min}$ denote the maximum and minimum *nonzero* eigenvalues of $\boldsymbol{A}\boldsymbol{A}^T$, respectively. For a fixed batch fraction, the optimal parameters $(\gamma_\lambda, \Delta_\lambda)$ of $\lambda_{\max}$ are

$$\begin{aligned} \gamma_\lambda &= \frac{1}{\zeta}\left(\frac{2}{\sqrt{\sigma^2_{\max}} + \sqrt{\sigma^2_{\min}}}\right)^2 \\ \text{and} \quad \Delta_\lambda &= \left(\frac{\sqrt{\sigma^2_{\max}} - \sqrt{\sigma^2_{\min}}}{\sqrt{\sigma^2_{\max}} + \sqrt{\sigma^2_{\min}}}\right)^2. \end{aligned} \tag{27}$$

In the full batch setting, i.e. $\zeta = 1$, these optimal parameters γ_λ and Δ_λ for $\lambda_{\max}$ are exactly the Polyak momentum parameters (20). Moreover, in this setting, there is no stochasticity so the Malthusian exponent disappears and the convergence rate (26) is $\lambda_{\max}$. We observe from (27) that for all fixed batch fractions, the optimal momentum parameter, Δ_λ, is independent of batch size. The only dependence on batch size appears in the learning rate. At first it appears that for small batch fractions, one can take large learning rates, but in that case, the inverse of the Malthusian exponent Ξ^{-1} dominates the convergence rate of SGD+M (26) and you cannot take γ and Δ to be as in (27) (See Figure 5).

We will define two subsets of parameter space: the *problem constrained regime* and the *algorithmically constrained regime* (or stochastically constrained regime). The problem constrained regime is for some tolerance $\varepsilon > 0$

$$\{(\gamma, \Delta) : 1 - \sqrt{\Xi} < (1 - \sqrt{\lambda_{\max}^{-1}})(1 - \varepsilon)\}. \tag{28}$$

The remainder we call the *algorithmically constrained* regime. To explain the tolerance: for finite n, it transpires that we always have $\Xi^{-1} > \lambda_{\max}$, but it could be vanishingly close to $\lambda_{\max}$ as a function of n. Hence we introduce the tolerance to give the correct qualitative behavior in finite n.

Proposition 4.5. *If the learning rate* $\gamma \leq \min(\frac{1+\Delta}{\zeta\sigma_{max}^2}, \frac{(1-\sqrt{\Delta})^2}{\zeta\sigma_{\min}^2})$, *with the trace condition* $\frac{8(1-\zeta)\gamma}{1-\Delta} \cdot \frac{1}{n} \operatorname{tr}(\boldsymbol{A}^T\boldsymbol{A}) < 1$, *then* (γ, Δ) *is in the problem constrained regime with* $\varepsilon = 1/2$.

Therefore by (26), we have that

$$\Psi_t - \Psi_\infty \leq D \left(\frac{4\lambda_{\max}}{(1 + \sqrt{\lambda_{\max}})^2} \right)^t, \quad \text{for some } D > 0; \tag{29}$$

we note that the expression in the parenthesis is $1 - \frac{1}{2}(1 - \lambda_{\max}) + \mathcal{O}((1 - \lambda_{\max})^2)$.

In the problem constrained regime, it is worthwhile to note that the overall convergence rate is the same as full batch momentum with adjusted learning rate, *i.e.*, the batch size does not play an important role as long as we are in the problem constrained regime.

4.4 Performance of SGD+M: implicit conditioning ratio (ICR)

An advantage of the exact loss trajectory is that we give a rigorous definition of the large batch and small batch regimes which reflect a transition in the convergence behavior of SGD+M. To do this we introduce the *condition number* κ, the *average condition number* $\bar{\kappa}$, and the *implicit conditioning ratio* (ICR) defined as

$$\bar{\kappa} \overset{\text{def}}{=} \frac{\frac{1}{n}\sum_{j \in [n]} \sigma_j^2}{\sigma_{\min}^2} < \frac{\sigma_{\max}^2}{\sigma_{\min}^2} \overset{\text{def}}{=} \kappa \quad \text{and} \quad \text{ICR} \overset{\text{def}}{=} \frac{\bar{\kappa}}{\sqrt{\kappa}}. \tag{30}$$

Here σ_j^2 are the eigenvalues of the Hessian of the least squares problem with $\sigma_{\max}^2$ and $\sigma_{\min}^2$ the largest and smallest (non-zero) eigenvalues. We refer to the *large batch* regime where $\zeta \geq$ ICR and the *small batch* regime where $\zeta \leq$ ICR.

We begin by giving a rate guarantee that holds in the problem constrained regime, for a specific choice of γ and Δ.

Proposition 4.6 (Good momentum parameters)**.** *Suppose the learning rate and momentum satisfy*

$$\gamma = \frac{(1-\sqrt{\Delta})^2}{\zeta\sigma_{\min}^2} \text{ and } \Delta = \max\left\{ \left(\frac{1 - \frac{\mathcal{C}}{\bar{\kappa}}}{1 + \frac{\mathcal{C}}{\bar{\kappa}}} \right), \left(\frac{1 - \frac{1}{\sqrt{2\kappa}}}{1 + \frac{1}{\sqrt{2\kappa}}} \right) \right\}^2, \quad \text{where } \mathcal{C} \overset{def}{=} \zeta/(8(1-\zeta)). \tag{31}$$

Then $\lambda_{\max} = \Delta$ *and for some* $C > 0$, *the convergence rate for SGD+M is*

$$\begin{aligned} \Psi_t - \Psi_\infty &\leq C \cdot \Delta^t \\ &= C \cdot \max\left\{ \left(\frac{1 - \frac{\mathcal{C}}{\bar{\kappa}}}{1 + \frac{\mathcal{C}}{\bar{\kappa}}} \right), \left(\frac{1 - \frac{1}{\sqrt{2\kappa}}}{1 + \frac{1}{\sqrt{2\kappa}}} \right) \right\}^{2t}. \end{aligned} \tag{32}$$

Remark 1. *We note that for all* Δ *satisfying* $\frac{(1-\sqrt{\Delta})^2}{\zeta\sigma_{\min}^2} \leq \frac{(1+\sqrt{\Delta})^2}{2\zeta\sigma_{\max}^2}$ *with the learning rate* γ *as in* (31), *we have that* $\lambda_{\max} = \Delta$. *By minimizing the* Δ *(i.e., by finding the fastest convergence rate), we get the formula for the momentum parameter in* (31).

The exact tradeoff in convergence rates (32) occurs when

$$\frac{\mathcal{C}}{\bar{\kappa}} = \frac{1}{\sqrt{2\kappa}}, \quad \text{or} \quad \zeta = \frac{\frac{8}{\sqrt{2}}\text{ICR}}{1 + \frac{8}{\sqrt{2}}\text{ICR}}. \tag{33}$$

As $\zeta \leq 1$, this condition is only nontrivial when ICR $\ll 1$, in which case $\zeta = \frac{8}{\sqrt{2}}$ICR, up to vanishing errors.

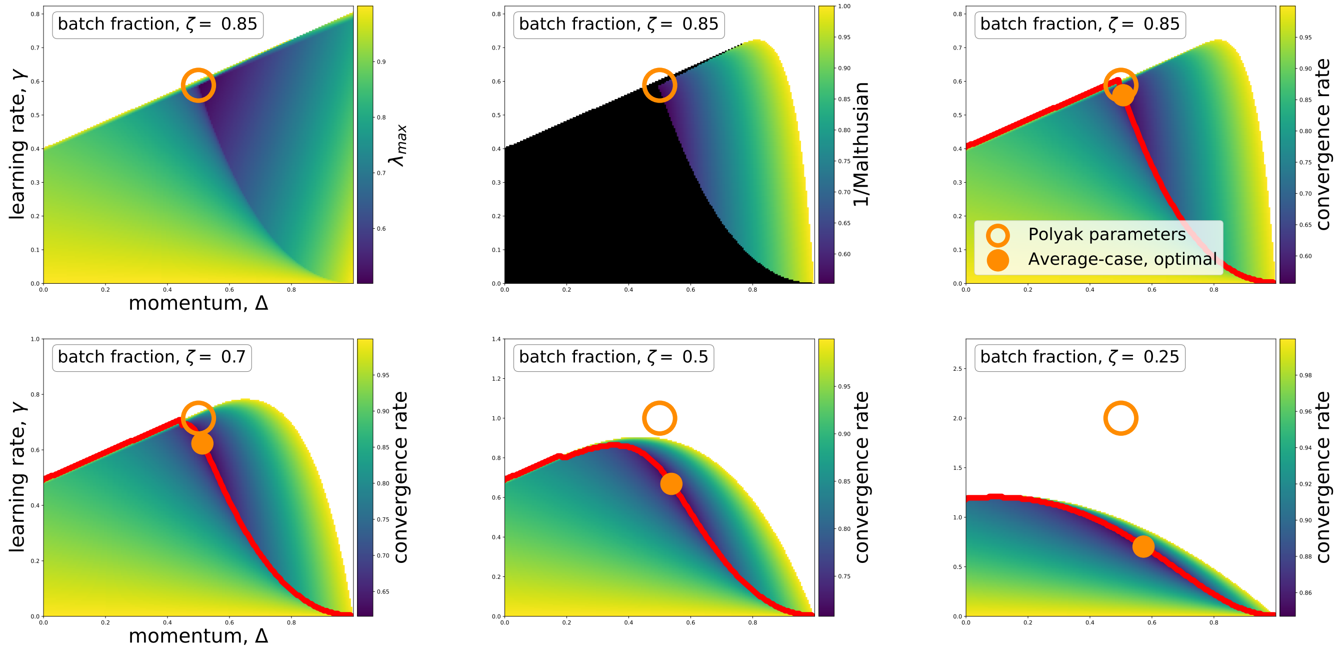

Figure 5: **Different convergence rate regions: problem constrained regime versus algorithmically constrained regime** for Gaussian random least squares problem with ($n = 2000 \times d = 1000$). Plots are functions of momentum (x-axis) and learning rate (y-axis). Analytic expression for $\lambda_{\max}$ (see (22)) – convergence rate of forcing term $F(t)$ – given in (top row, column 1) represents the problem constrained region. (top row, column 2) plots 1/(Malthusian exponent) ((25)); black region is where the Malthusian exponent Ξ does not exist. This represents the algorithmically constrained region. Finally, (top row, column 3 and bottom row) plots convergence rate of SGD+M $= \max\{\lambda_{\max}, \Xi^{-1}\}$, (see (26)), for various batch fractions. When the Malthusian exponent does not exist (black), $\lambda_{\max}$ takes over the convergence rate of SGD+M; otherwise the noise in the algorithm (i.e. Malthusian exponent Ξ) dominates. Optimal parameters that maximize $\lambda_{\max}$ denoted by Polyak parameters (orange circle, (27)) and the optimal parameters for SGD+M (orange dot); below red line is the problem constrained region; otherwise the algorithmic constrained region. When batch fractions $\zeta = 0.85$ and $\zeta = 0.7$ (top row and bottom row, column 1) (i.e., large batch), the SGD+M convergence rate is the deterministic momentum rate of $1/\sqrt{\kappa}$. As the batch fraction decreases ($\zeta = 0.25$), the convergence rate becomes that of SGD and the optimal parameters of SGD+M and Polyak parameters are quite far from each other. The Malthusian exponent (algorithmically constrained region) starts to control the SGD+M rate as batch fraction → 0.

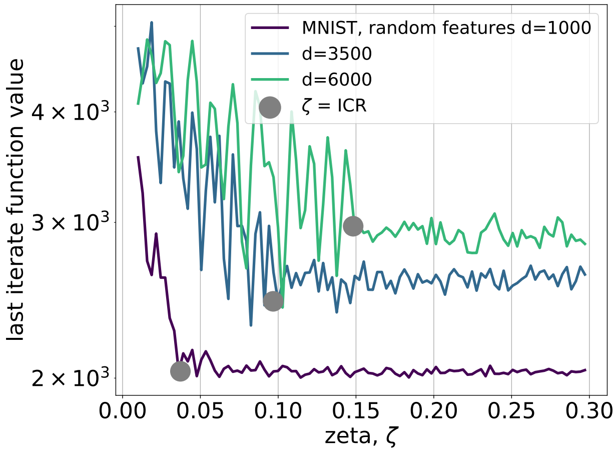

Figure 6: **ICR and batch saturation on MNIST data.** SGD with momentum using a batch fraction ζ on MNIST data [33]; training loss is given after 20 iterations. Increasing the batch size yields proportional complexity improvements up to a saturation point (gray dot, explicit formula in [34]) which occurs before the full gradient is deployed. This yields the first provable optimal linear rate for stochastic momentum learning algorithm that matches its deterministic equivalent.

Large batch ($\zeta \geq$ ICR). In this regime SGD+M's performance matches the performance of the heavy-ball algorithm with the Polyak momentum parameters (up to absolute constants). More specifically, with the choices of γ and Δ in Proposition 4.6, the linear rate of convergence of SGD+M is $1-\frac{c}{\sqrt{\kappa}}$ for an absolute c. Note that ζ does not appear in the rate, and in particular there is no gain in convergence rate by increasing the batch fraction.

Small batch ($\zeta \leq$ ICR). In the small batch regime, the value of $\mathcal{C}$ is relatively small and the first term is dominant in (32), and so the linear rate of convergence of SGD+M is $1-\frac{c\zeta}{\overline{\kappa}}$ for some absolute constant $c > 0$. In this regime, there is a benefit in increasing the batch fraction, and the rate increases linearly with the fraction. We note that on expanding the choice of constants in small ζ the choices made in Proposition 4.6 are

$$\Delta \approx 1-\frac{\zeta}{8\overline{\kappa}} \quad \text{and} \quad \gamma \approx \frac{\zeta}{256\overline{\kappa}^2\sigma_{\min}^2}.$$

This rate can also be achieved by taking $\Delta = 0$, i.e. mini-batch SGD with no momentum. Moreover, it is not possible to beat this by using momentum; we show the following lower bound:

Proposition 4.7. *If $\zeta \leq \min\{\frac{1}{2}, ICR\}$ then there is an absolute constant $C > 0$ so that for convergent (γ,Δ) (those satisfying Proposition 4.3), $\sqrt{\lambda_{\max}} \geq 1-\frac{C\zeta}{\overline{\kappa}}$.*

This is a lower bound on the rate of convergence by Theorem 4.4.

Conclusions. While the engineering side of machine learning has leapt ahead, the theoretical explanation for what is happening in ML training has largely been left behind. The needed theory of optimization to close this gap should fit 3 key aspects: (1) the algorithm is a gradient-based method, (2) the training loss is a high–dimensional "finite-sum", and (3) the model is "the right type" of nonconvex problem.

In this work, we presented a theory that does 2 of the 3; we outlined a framework for addressing this gap between theory and practice by incorporating a deterministic resolvent condition into the assumptions. For the ℓ^2-regularized least squares problem, the stochastic learning algorithms concentrate around a simple, predictable path. By analyzing this path, one can draw insights into average-case complexity and parameter selection properties, all of which have enormous practical implications for making machine learning work.

Clearly the most urgent direction of future research is away from the least squares setting, to handle more general losses and some types of nonconvexity. On the one hand, there is evidence that the right type of nonconvex problems are not so far from convex, taking to heart that, for example, wide neural networks degenerate to kernel regression problems, which are covered by this framework. On the other hand, as we move away from the least squares setting, we truly do not know what we do not know; there are many other important model problems which need the high-dimensional optimization treatment, such as generalized linear models, inverse problems like phase retrieval, and neural networks.

References

[1] B. Adlam and J. Pennington. The Neural Tangent Kernel in High Dimensions: Triple De-

scent and a Multi-Scale Theory of Generalization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 74–84. PMLR, 13–18 Jul 2020.

[2] B. Adlam and J. Pennington. Understanding Double Descent Requires A Fine-Grained Bias-Variance Decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 11022–11032, 2020.

[3] Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

[4] Y. Arjevani, Y. Carmon, and J.C. Duchi. Lower bounds for non-convex stochastic optimization. *Math. Program.*, 155, 2022.

[5] S. Asmussen. *Applied probability and queues*, volume 51 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, second edition, 2003. Stochastic Modelling and Applied Probability.

[6] S. Asmussen. *Applied probability and queues*, volume 51 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, second edition, 2003. Stochastic Modelling and Applied Probability.

[7] F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, 2013.

[8] M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proc. Natl. Acad. Sci. USA*, 116(32):15849–15854, 2019.

[9] G. Ben Arous, R. Gheissari, and A. Jagannath. High-dimensional limit theorems for sgd: Effective dynamics and critical scaling. *arXiv preprint arXiv:2206.04030*, 2022.

[10] D. Bertsekas. A new class of incremental gradient methods for least squares problems. *SIAM J. Optim.*, 7(4):913–926, 1997.

[11] D. Bertsekas and J. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM J. Optim.*, 10(3):627–642, 2000.

[12] J. Blanchard, C. Cartis, and J. Tanner. Compressed sensing: how sharp is the restricted isometry property? *SIAM Rev.*, 53(1):105–125, 2011.

[13] R. Bollapragada, T. Chen, and R. Ward. On the fast convergence of minibatch heavy ball momentum. *arXiv preprint arXiv:2206.07553*, 2022.

[14] B. Bordelon and C. Pehlevan. Learning Curves for SGD on Structured Features. In *International Conference on Learning Representations (ICLR)*, 2022.

[15] L. Bottou, F.E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

[16] B. Can and M. Gurbuzbalaban. Entropic risk-averse generalized momentum methods. *arXiv preprint arXiv:2204.11292*, 2022.

[17] M. Celentano, C. Cheng, and A. Montanari. The high-dimensional asymptotics of first order methods with random data. *arXiv preprint arXiv:2112.07572*, 2021.

[18] K. Chandrasekher, A. Pananjady, and C. Thrampoulidis. Sharp global convergence guarantees for iterative nonconvex optimization: A gaussian process perspective. *arXiv preprint arXiv:2109.09859*, 2021.

[19] D. Davis and D. Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM J. Optim.*, 29(1):207–239, 2019.

[20] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[21] A. Defossez and F. Bach. Averaged Least-Mean-Squares: Bias-Variance Trade-offs and Optimal Sampling Distributions. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*,

volume 38 of *Proceedings of Machine Learning Research*, pages 205–213, 2015.

[22] E. Dobriban and S. Wager. High-dimensional asymptotics of prediction: ridge regression and classification. *Ann. Statist.*, 46(1):247–279, 2018.

[23] Y. Drori and O. Shamir. The complexity of finding stationary points with stochastic gradient descent. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 2658–2667. PMLR, 13–18 Jul 2020.

[24] C. Fang, C. J. Li, Z. Lin, and T. Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[25] S. Ghadimi and G. Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.*, 23(4):2341–2368, 2013.

[26] R. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik. SGD: General analysis and improved rates. In *International Conference on Machine Learning (ICML)*. PMLR, 2019.

[27] G. Gripenberg. On the resolvents of nonconvolution Volterra kernels. *Funkcial. Ekvac.*, 23(1):83–95, 1980.

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

[29] P. Jain, S. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford. Parallelizing Stochastic Gradient Descent for Least Squares Regression: Mini-batching, Averaging, and Model Misspecification. *Journal of Machine Learning Research*, 18(223):1–42, 2018.

[30] P. Jain, S. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford. Accelerating Stochastic Gradient Descent for Least Squares Regression. In *Proceedings of the 31st Conference On Learning Theory (COLT)*, volume 75 of *Proceedings of Machine Learning Research*, pages 545–604. PMLR, 2018.

[31] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013.

[32] R. Kidambi, P. Netrapalli, P. Jain, and S. Kakade. On the insufficiency of existing momentum schemes for stochastic optimization. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9, 2018.

[33] Y. LeCun, C. Cortes, and C. Burges. ”mnist” handwritten digit database, 2010.

[34] K. Lee, A.N. Cheng, E. Paquette, and C. Paquette. Trajectory of Mini-batch Momentum: Batch Size Saturation and Convergence in High-dimensions. *arXiv preprint arXiv:2206.01029*, 2022.

[35] Z. Liao, R. Couillet, and M. Mahoney. A Random Matrix Analysis of Random Fourier Features: Beyond the Gaussian Kernel, a Precise Phase Transition, and the Corresponding Double Descent. *arXiv preprint arXiv:2006.05013*, 2020.

[36] C. Liu and M. Belkin. Accelerating SGD with momentum for over-parameterized learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[37] S. Mei and A. Montanari. The generalization error of random features regression: precise asymptotics and the double descent curve. *Comm. Pure Appl. Math.*, 75(4):667–766, 2022.

[38] F. Mignacco, F. Krzakala, P. Urbani, and L. Zdeborova. Dynamical mean-field theory for stochastic gradient descent in gaussian mixture classification. *arXiv preprint arXiv:2006.06098*, 2000.

[39] E. Moulines and F. Bach. Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24, 2011.

[40] P. Nakkiran, B. Neyshabur, and H. Sedghi. The Deep Bootstrap Framework: Good Online

Learners are Good Offline Generalizers. In *International Conference on Learning Representations (ICLR)*, 2021.

[41] D. Needell, N. Srebro, and R. Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Math. Program.*, 155(1-2, Ser. A):549–573, 2016.

[42] A. S. Nemirovsky and D. B. and Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Inc., New York, 1983. Translated from the Russian and with a preface by E. R. Dawson.

[43] S. Oymak, B. Recht, and M. Soltanolkotabi. Sharp time-data tradeoffs for linear inverse problems. *IEEE Trans. Inform. Theory*, 64(6):4129–4158, 2018.

[44] C. Paquette, K. Lee, F. Pedregosa, and E. Paquette. SGD in the Large: Average-case Analysis, Asymptotics, and Stepsize Criticality. *arXiv preprint arXiv:2102.04396*, 2021.

[45] C. Paquette and E. Paquette. Dynamics of Stochastic Momentum Methods on Large-scale, Quadratic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

[46] E. Paquette, C. Paquette, B. Adlam, and J. Pennington. Homogenization of SGD in high-dimensions: exact dynamics and generalization properties. *arXiv preprint arXiv:2205.07069*, 2022.

[47] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 04, 1964.

[48] H. Robbins and S. Monro. A Stochastic Approximation Method. *Ann. Math. Statist.*, 1951.

[49] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 2017.

[50] P. Sur and E. Candès. A modern maximum-likelihood theory for high-dimensional logistic regression. *Proc. Natl. Acad. Sci. USA*, 116(29):14516–14525, 2019.

[51] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML)*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 2013. PMLR.

[52] N. Tripuraneni, B. Adlam, and J. Pennington. Covariate Shift in High-Dimensional Random Feature Regression. *arXiv preprint arXiv:2111.08234*, 2021.