SGD WITH LARGE STEP SIZES LEARNS SPARSE FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

We showcase important features of the dynamics of the Stochastic Gradient Descent (SGD) in the training of neural networks. We present empirical observations that the commonly used large step sizes (i) lead the iterates to jump from one side of a valley to the other causing *loss stabilisation* (ii) this stabilisation induces a hidden stochastic dynamics orthogonal to the bouncing directions that *biases it implicitly* toward simple predictors. Furthermore, we show empirically that the longer large step sizes keep SGD high in the loss landscape valleys, the better the implicit regularization can operate and find sparse representations. Notably, no explicit regularization is used so that the regularization effect comes solely from the SGD training dynamics influenced by the step size schedule. Therefore, these observations unveil how, through the step size schedules, both gradient and noise drive together the SGD dynamics through the loss landscape of neural networks. We justify these findings theoretically through the study of simple neural network models. Finally, we shed a new light on some common practice and observed phenomena when training neural networks.

1 INTRODUCTION

During the last decade, deep neural networks have accomplished remarkable achievements on a wide variety of tasks. Yet, the understanding of their remarkable effectiveness remains incomplete. From an optimization perspective, stochastic training procedures challenge every knowledge theoreticians have drawn from convex models. In particular, large step-size schedules used in practice lead to unexpected patterns of stabilization and sudden drops in the training loss profile. From a generalization perspective, overparametrized deep networks generalize well while fitting perfectly the data and without any explicit regularization (Zhang et al., 2016). This surprising fact implies that optimization and generalization are tightly intertwined: neural networks find well generalizing solutions thanks to the optimization algorithm. This property, known as *implicit bias* or *algorithmic regularization*, has been studied recently both in the case of regression (Soudry et al., 2018; Woodworth et al., 2020) and classification (Lyu and Li, 2020; Chizat and Bach, 2020; Ji and Telgarsky, 2020). However, the time to enter the sparse feature learning regimes is overlong in these theoretical results and they do not capture the practical training setup.

In this paper, we aim at staying closer to the experimental practice. Specifically, we propose to understand stochastic gradient descent (SGD) with the *large step-size* schedule introduced in the original ResNet paper (He et al., 2016) where the step size is first kept constant to a large value and then decayed (potentially, multiple times). We illustrate this behavior in Fig. 1 where we reproduce a minimal setting with no data augmentation, momentum, and only one decrease of the step size. We draw attention to two important observations (a) the loss stays approximately constant when the step size is large and (b) despite no progress on the training loss, running this phase for longer leads to better generalization. We refer to this large step-size phase as *loss stabilization*. Our main contribution is to unveil the hidden dynamics behind this phase: loss stabilization fosters the exploration of SGD in the loss landscape and enables the model to learn *sparse features*.

1.1 OUR CONTRIBUTION

SGD is GD + specific label noise. Noise seems to play a role of paramount importance to help the training dynamics travel the loss landscape (Keskar et al., 2016; Xing et al., 2018). There have been attempts to highlight this property by studying stochastic differential equations (SDEs), but this

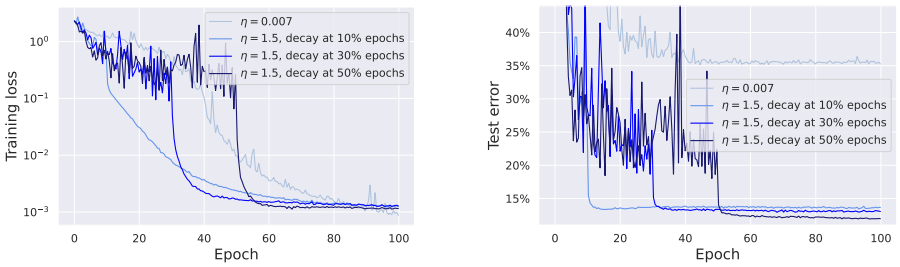

Figure 1: A typical training dynamics for a ResNet-18 trained on CIFAR-10. We use weight decay but no momentum or data augmentation for this experiment. We see a substantial difference in generalization (as large as 12% vs. 36% test error) depending on the step size and its schedule. When the training loss stabilizes, there is a hidden progress occurring which we aim to characterize.

approach has not been completely satisfactory since the noise covariance is rarely well understood. We show here that SGD can be cast exactly as GD with a specific label noise that scales with the training loss. This allows to unveil the specific shape and scale of the SGD noise allowing to model it in a suitable SDE.

Large step sizes SGD enables exploration by loss stabilization. On the one hand, we show that loss stabilization is not due to a saddle-to-saddle behavior (Jacot et al., 2021; Barak et al., 2022) but rather to bouncing side-to-side above some valley and where sudden drops occur if the step size is lowered. We model it as a fast-slow dynamics where (i) the fast movement is determined by the bouncing directions causing loss stabilization (ii) the slow dynamics drives SGD to explore efficiently the loss landscape by the combination of the gradient and multiplicative noise.

Exploration leads SGD to learn sparse features. The hidden slow dynamics during loss stabilization is showcased as a SDE whose multiplicative noise is related to the neural tangent kernel features. We show in a toy model (diagonal linear networks) that this multiplicative noise structure leads to learn sparse features. Building on this, we conjecture a similar effect for more complicated architectures and experimentally show it on neural nets of increasing complexities. We exhibit how the loss stabilization phase is crucial to make sure that convergence does not occur too early to suboptimal near predictors; hence showing the benefit from the hidden dynamics of SGD.

Insights from our understanding. Overall, we observe a clear general picture: the hidden optimization dynamics induced by large step sizes and loss stabilization allows to transition from a partially lazy training regime to the sparse feature learning regime. We shed a new light on phenomena observed in practice and give a sense on when the exploration and fitting phases happen. We argue that after a warm up phase, SGD *first* tends to learn sparse feature of the signal *and then eventually* fits it when the step size is decayed enough. Finally, we discuss informally how many deep learning regularization methods (weight decay, BatchNorm, SAM) also fit into the same picture.

1.2 Related work

He et al. (2016) popularized the piecewise constant step-size schedule which often exhibits a clear loss stabilization pattern. However, they did not provide any explanations for such training dynamics and its implicit regularization effect. Non-monotonic patterns of the training loss have been explored in recent works. We emphasize that the loss stabilization we consider is different (i) from the catapult mechanism (Lewkowycz et al., 2020) as there the training loss does not stabilize, and (ii) from the unstable convergence regime described in the edge of stability paper (Cohen et al., 2021) as we consider an even larger step-size regime.

Past works conjectured that large step sizes correspond to minimization of some hidden complexity measures related to flatness of minima (Keskar et al., 2016; Smith and Le, 2017). Notably, Xing et al. (2018) point out that SGD moves through the loss landscape bouncing between the walls of a valley where the role of the step size is to guide exploration of SGD towards a flatter minimum. However, many typically used flatness definitions are questionable for this purpose (Dinh et al., 2017) and there is no clear conclusion about the hidden complexity measure and its properties. For example, full-batch gradient descent with large step sizes (unlike SGD) leads to flat solutions which

are not well-generalizing (Kaur et al., 2022). It is possible to bridge the gap between GD and SGD by using explicit regularization methods as in Geiping et al. (2021). We instead focus on the implicit regularization of SGD which remains the most practical approach for training deep networks.

The importance of large step sizes has been investigated with diverse motivations. However, we believe that existing approaches do not sufficiently capture the *hidden dynamics* behind the loss stabilization phenomenon observed for deep networks. Attempts to explain it on convex models (Nakkiran, 2020; Wu et al., 2021; Beugnot et al., 2022) are inherently incomplete since it is an non-convex phenomenon related to the existence of many zero solutions with very different generalization properties. Li et al. (2019b) analyzed the role of loss stabilization for a synthetic distribution containing different patterns, but it is not clear how this analysis can be extended to general problems. Works based on stability analysis characterize the properties of the minimum that SGD or GD can potentially converge depending on the step size (Wu et al., 2018; Mulayoff et al., 2021; Ma and Ying, 2021; Nacson et al., 2022) but without capturing the entire training dynamics.

Finally, SGD with label noise has been studied because of its beneficial regularization effect and its resemblance to SGD's standard noise. Its implicit bias has been first characterized by Blanc et al. (2020) and extended by Li et al. (2021). However, their analysis only holds in the final phase of the training, close to a zero-loss manifold. Our work instead is closer in spirit to Pillaud-Vivien et al. (2022) where the label noise dynamics is analyzed in the *central* phase of the training.

2 The effective dynamics of SGD with large step-size: sparse feature learning

In this section, we show that large step sizes lead the loss to stabilize through the bouncing of the SGD iterates and unveil the effective dynamics induced by this loss stabilization. To clarify our exposition we showcase our results for the mean square error but other losses like the cross-entropy carry the same properties (Wojtowytsch, 2021b). We consider a parameterized family of prediction functions $\mathcal{H} := \{x \to h_\theta(x),\ \theta \in \mathbb{R}^p\}$, a setting which encompasses neural networks. In our case, the train loss on input/output samples $(x_i, y_i)_{1 \le i \le n} \in \mathbb{R}^d \times \mathbb{R}$ reads

$$\mathcal{L}(\theta) := \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(x_i) - y_i\right)^2 . \tag{1}$$

We consider the overparameterized setting, i.e. $p \gg n$, hence, there shall exists many parameters θ^* that lead to zero loss, i.e. interpolate fully the dataset. Therefore, the question of to which interpolator the algorithm converges is of paramount importance in terms of generalization: it is referred as the *implicit bias* of the algorithm. As explained above, we focus on the SGD recursion with step size $\eta > 0$, initialized at $\theta_0 \in \mathbb{R}^p$ and which writes for all $t \in \mathbb{N}$,

$$\theta_{t+1} = \theta_t - \eta(h_{\theta_t}(x_{i_t}) - y_{i_t})\nabla_\theta h_{\theta_t}(x_{i_t}), \tag{2}$$

where $i_t \sim \mathcal{U}\left([\![1, n]\!]\right)$ is the uniform distribution over the samples. We note that considering SGD with mini-batch would lead to similar analysis and conclusions.

2.1 SGD is GD with specific label noise

To put emphasis on the combined roles of gradient and noise, we draw the connection between the SGD dynamics and that of full batch GD where we add a specific noise to the labels at each iteration.

Proposition 1. *Let $(\theta_t)_{t \ge 0}$ follows the SGD dynamics Eq.(2) with sampling function $(i_t)_{t \ge 0}$. Define for each $t \ge 0$, the random vector $\xi_t \in \mathbb{R}^n$ such that for all $i \in [\![1, n]\!]$,*

$$[\xi_t]_i := (h_{\theta_t}(x_i) - y_i)(1 - n\mathbf{1}_{i = i_t}). \tag{3}$$

Then, $(\theta_t)_{t \ge 0}$ follows the dynamics of the full batch gradient dynamics on $\mathcal{L}$ where at each iteration $t \ge 0$ we inject noise ξ_t to the label vector $y \in \mathbb{R}^n$: that is at step $t \ge 0$, we compute the gradient of $\mathcal{L}$ considering the random labels $y_t := y + \xi_t$.

This proposition links the dynamics of SGD to the one of GD plus a certain amount of label noise. The scale of this induced noise is characterized in the following lemma. Refer App. A for the proofs.

Lemma 2. *For $t \geq 0$, let $\xi_t \in \mathbb{R}^n$ be defined as in Proposition 1, then ξ_t is a mean zero random vector with variance such that $\frac{1}{n(n-1)}\mathbb{E}\left\|\xi_t\right\|^2 = 2\mathcal{L}(\theta_t)$.*

Hence, the effective label noise decays with the training loss. The effect of the SGD noise decreases therefore during the optimization process. However, we highlight in the following section that loss can stabilize because of large step sizes. The amount of label noise becomes constant. This feature enables to unveil the effective dynamics that take place during loss stabilization.

2.2 The effective dynamics behind loss stabilization

On loss stabilization. For generic quadratic costs, e.g. $F(\beta) := \|X\beta - y\|^2$, gradient descent with step size η is convergent for $\eta < 2/\lambda_{\max}$, divergent for $\eta > 2/\lambda_{\max}$ and converge to a bouncing 2-periodic dynamics for $\eta = 2/\lambda_{\max}$, where $\lambda_{\max}$ is the largest eigenvalue of the Hessian. However, the practitioner is not likely to hit perfectly this unstable step size, hence almost surely, the dynamics shall either converge or diverge. Yet, non-quadratic costs bring to this picture a particular complexity: it has been shown that, even for non-convex toy models, there exist an open interval of step sizes for which the gradient descent neither converge nor diverge (Ma et al., 2022; Chen and Bruna, 2022). As we are interested in SGD, we complement this result by presenting a toy example in which *loss stabilization* occurs almost surely *in the case stochastic updates*. Indeed, consider a regression problem with quadratic parameterization on one-dimensional data inputs x_i's, coming from a distribution $\hat{\rho}$, and outputs generated by the linear model $y_i = x_i\theta_*^2$. The loss writes $F(\theta) := \frac{1}{4}\mathbb{E}_{\hat{\rho}}\left(y - x\theta^2\right)^2$, and the SGD iterates with step size $\eta > 0$ follow, for any $t \in \mathbb{N}$,

$$\theta_{t+1} = \theta_t + \eta\,\theta_t\,x_{i_t}\left(y_{i_t} - x_{i_t}\theta_t^2\right) \qquad \text{where} \qquad x_{i_t} \sim \hat{\rho}. \tag{4}$$

For the sake of concreteness and clarity, suppose that $\theta_* = 1$ and $\mathrm{supp}(\hat{\rho}) = [a, b]$, we have the following proposition (a more general result can be found in Proposition 7 in Appendix).

Proposition 3. *For any $\eta \in (a^{-2}, 1.25 \cdot b^{-2})$ and initialization $\theta_0 \in (0, 1)$, for $t > 0$,*

$$\delta_1 < F(\theta_t) < \delta_2 \qquad \textit{almost surely, and} \tag{5}$$

$$\exists T > 0, \forall k > T, \qquad \theta_{t+2k} < 1 < \theta_{t+2k+1} \qquad \textit{almost surely.} \tag{6}$$

where $\delta_1, \delta_2, T > 0$ are constant given in the Appendix.

The proposition is divided in two parts: for some range of step sizes, Eq.(5) shows that the loss stabilizes in between level sets δ_1 and δ_2 and Eq.(6) shows that after some initial phase, the iterates bounce from one side of the *loss valley* to the other one. Note that despite stochasticity of the procedure, the results hold *almost surely*.

The effective dynamics. As observed in the prototypical SGD training dynamics of Fig. 1 and proved in the non-convex toy model of Proposition 3, large step sizes lead the loss to stabilize around some level set. To further understand the effect of this loss stabilization in parameter space, we shall assume perfect stabilization and thus that the loss $L(\theta_t)$ remains equal to some constant. Then, from Proposition 1, that links SGD with GD plus label noise, and Lemma 2, that shows that the noise scales like the loss, we infer the following claim.

Claim 4. *During loss stabilization, SGD is well modelled by GD with constant label noise.*

Label noise dynamics have been studied recently (Blanc et al., 2020; Damian et al., 2021; Li et al., 2021) thanks to their connection with continuous time Stochastic Differential Equations (SDEs). To properly write a SDE model, the drift should match the gradient descent and the noise should have the correct covariance structure (Li et al., 2019a; Wojtowytsch, 2021a). Claim 4 implies that the noise at state θ is spanned by the gradient vectors $\{\nabla_\theta h_\theta(x_1), \ldots, \nabla_\theta h_\theta(x_n)\}$ and has a constant intensity corresponding to the loss stabilization at a level $\delta > 0$. We propose the following SDE model

$$\mathrm{d}\theta_t = -\nabla_\theta \mathcal{L}(\theta_t)\mathrm{d}t + \sqrt{\eta\delta}\,\phi_{\theta_t}(X)^\top \mathrm{d}B_t, \tag{7}$$

where $\phi_\theta(X) := [\nabla_\theta h_\theta(x_i)^\top]_{i=1}^n \in \mathbb{R}^{n\times p}$ is the *noise feature matrix* and $(B_t)_{t\geq 0}$ is a standard Brownian motion in $\mathbb{R}^n$. Remarkably, the features $\phi_\theta(X)$ are the one given by the Neural Tangent Kernel (NTK) (Jacot et al., 2018) at state θ, this is the reason why we shall also name it this way. This SDE can be seen as *the effective slow dynamics* that drives the iterates while they bounce *rapidly* in some directions at the level set δ (fast dynamics). It highlights the combination of the deterministic part of the full batch gradient and the noise induced by SGD at level set δ. In the next section, we leverage the SDE (7) to understand the implicit bias of such learning dynamics.

2.3 SPARSE FEATURE LEARNING

In this section, we give insights on the effective dynamics given by Eq.(7). We begin with a simple model, diagonal linear networks, that showcase a sparse-inducing dynamics and further disclose our general message about the overall implicit bias promoted by the effective dynamics.

2.3.1 A WARM-UP: DIAGONAL LINEAR NETWORKS

An appealing example of simple non-linear networks that help in forging an intuition for more complicated architectures are diagonal linear networks (Vaskevicius et al., 2019; Woodworth et al., 2020; HaoChen et al., 2020; Pesme et al., 2021). They are two-layer linear networks with only diagonal connections: the prediction function writes $h_{u,v}(x) = \langle u, v \odot x\rangle = \langle u \odot v, x\rangle$ where $\odot$ denotes *elementwise* multiplication. Even though the loss is convex in the associated linear predictor $\beta := u \odot v \in \mathbb{R}^d$, it is not in (u, v), hence the training of such simple models already exhibit a rich non-convex dynamics. In this case, $\nabla_u h_{u,v}(x) = v \odot x$, and the SDE model Eq.(7) writes

$$\mathrm{d}u_t = -\frac{1}{n}\left[X^\top(X(u_t \odot v_t) - y)\right] \odot v_t \,\mathrm{d}t + \sqrt{\eta\delta}\, v_t \odot \left[X^\top \mathrm{d}B_t\right], \tag{8}$$

where $(B_t)_{t\geq 0}$ is a standard Brownian motion in $\mathbb{R}^n$. Equations are symmetric for $(v_t)_{t\geq 0}$.

What is the behaviour of this effective dynamics? Pillaud-Vivien et al. (2022) answered this question by analyzing a similar stochastic dynamics and unveiled the sparse nature of the solutions retrieved. Indeed, if we denote β^* the sparsest linear predictor that interpolates the data, it is shown that the associated linear predictor $\beta_t = u_t \odot v_t$: (i) converges exponentially fast to zero outside of the support of β^* (ii) is *with high probability* in a $\mathcal{O}(\sqrt{\eta\delta})$ neighborhood of β^* in its support.

Overall conclusion on the model. During a first phase, SGD with large step sizes η decrease the train loss until stabilization at some level set $\delta > 0$. During this loss stabilization, an effective noise-driven dynamics takes place. It shrinks the coordinates outside of the support of the sparsest signal and oscillates in parameter space at level $\mathcal{O}(\sqrt{\eta\delta})$ on its support. Then, decreasing the step size leads to perfect recovery of the sparsest predictor. This behaviour is illustrated in Figure 2.

2.3.2 THE SPARSE FEATURE LEARNING CONJECTURE FOR MORE GENERAL MODELS

Results for diagonal linear nets recalled in the previous paragraph show that the noisy dynamics Eq.(8) induce a *sparsity bias*. As emphasized in HaoChen et al. (2020), this effect is largely due to the multiplicative structure of the noise $v \odot [X^\top \mathrm{d}B_t]$ that, in this case, has a shrinking effect *on the coordinates* (because of the coordinate-wise multiplication with v). In the general case, we see, thanks to Eq.(7), that the same multiplicative structure of the noise still happens but this time *with respect to the noise feature matrix* $\phi_\theta(X)$. Hence, this leads to think that, similarly to the diagonal linear network case, the implicit bias of the noise is that of a shrinkage effect applied to $\phi_\theta(X)$. Remarkably, from a stability perspective, Blanc et al. (2020) showed that the SGD dynamics implicitly tries to minimize the *Frobenius norm* $\|\phi_\theta(X)\|_{\mathrm{F}} = \sum_{i=1}^n \|\nabla_\theta h_\theta(x_i)\|^2$. However, building on (Pillaud-Vivien et al., 2022), when noise is too high, the multiplicative factor has a stronger effect: it drives each direction to the lowest norm. Indeed, an interesting property of Brownian motion is that, for $v \in \mathbb{R}^p$, $\langle v, B_t\rangle = \|v\|_2 W_t$ in law and where (W_t) is a one-dimensional Brownian motion. Hence,

The noise part of Eq.(7) tries to minimize the ℓ_2-norm of ***each*** *column of $\phi_\theta(X)$.*

Note that we put **each** in emphasis to stress on the difference with the sum structure of the Frobenius norm. Keep also in mind that the *fitting part* of the dynamics prevents the NTK feature matrix to collapse to zero, but as soon as they are not needed to fit the signal, *columns* can be reduced to zero. We provide below a specification of this implicit bias for different architectures:

- **Diagonal linear networks:** $h_{u,v}(x) = \langle u \odot v, x\rangle$, then $\nabla_{u,v} h_{u,v}(x) = [v \odot x, u \odot x]$. For generic data X, minimizing the norm of each column of $\phi_{u,v}(X) = [X \odot \mathbf{1}v^\top, X \odot \mathbf{1}u^\top]^\top$ amounts to put the maximum of zeros coordinate and hence to minimize $\|u \odot v\|_0$.
- **ReLU networks:** We take the prototypical one hidden layer to exhibit the sparsification effect. Let $h_{a,W}(x) = \langle a, \sigma(Wx)\rangle$, then $\nabla_a h_{a,W}(x) = \sigma(Wx)$ and $\nabla_{w_j} h_{a,W}(x) = a_j x \mathbf{1}_{\langle w_j, x\rangle > 0}$. Note that the ℓ_2-norm of the column resp. to the neuron is reduced when it is activated at a *minimal number of training points*, hence the implicit bias enables the learning of *sparse data-active features*.

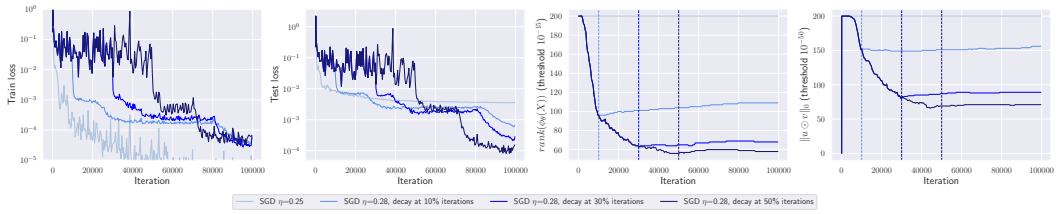

Figure 2: Diagonal linear networks. We observe loss stabilization, better generalization for longer schedules, minimization of the rank of $\phi_\theta(X)$ and sparsity of the predictor $u \odot v$.

Finally, note that when some directions are needed to fit the data, similarly activated neurons align to fit, allowing the rank of $\phi_\theta(X)$ to be also a good proxy for this feature sparsity.

Overall, fully understanding theoretically the structural implications of the implicit bias described here is an exciting route for future work. We show next that the conjectured sparsity indeed occurs empirically, as well as the rank reduction used as a good proxy.

3 Empirical evidence of sparse feature learning driven by SGD

Here we present empirical results for neural networks of increasing complexity: from diagonal linear networks to deep ReLU networks on CIFAR-10. We make the following common observations for all these networks trained using SGD schedules with large step sizes:

(O1) Loss stabilization: training loss stabilizes around a high level set until the step size is decayed,

(O2) Generalization benefit: longer loss stabilization leads to better generalization,

(O3) Sparse feature learning: longer loss stabilization leads to sparser features.

Importantly, *we use no explicit regularization* in our experiments so that the training dynamics is driven purely by SGD and the step size schedule. Additionally, in some cases, we cannot find step sizes that would lead to loss stabilization. In such cases, whenever explicitly mentioned, we use a *warmup* step size schedule—i.e., increasing step sizes according to some schedule—to make sure that the training loss stabilizes around some level set. Such warmup schedules are commonly used in practice (He et al., 2016; Devlin et al., 2018). Warmup is often motivated purely from the optimization perspective as a way to accelerate training (Agarwal et al., 2021) but we interpret it as a way to enhance the implicit regularization effect of SGD.

Measuring sparse feature learning. Our main insight is that the noise feature matrix is significantly simplified in the loss stabilization phase, and that the rank of $\phi_\theta(X)$ is a good proxy to track this phenomenon. We compute it over iterations for each model (except deep networks where it is not feasible) by using a fixed threshold on the singular values of $\phi_\theta(X)$ normalized by the largest singular value. In this way, we ensure that the difference in the rank that we detect is not simply due to a different scale of $\phi_\theta(X)$. Moreover, we always compute $\phi_\theta(X)$ on the number of fresh samples equal to the number of parameters $|\theta|$ to make sure that rank deficiency is not coming from a small value of $n \ll |\theta|$ which is the case in the overparametrized settings we consider.

Furthermore, we wanted to also exhibit a direct structural measure showcasing the sparsity of the features. This motivates us to count the average number of *distinct* (i.e., counting a group of highly correlated activations as one), *non-zero* activations across the network which we refer to as the *feature sparsity coefficient*. Unlike rank$(\phi_\theta(X))$, this coefficient scales to deep networks and has an easy-to-grasp meaning. Moreover, feature sparsity especially at the last layers is desirable from the interpretability perspective (Zeiler and Fergus, 2014) as well as for transfer learning since it signifies that only a few high-level patterns are detected by the network.

3.1 Sparse feature learning in diagonal linear networks

Setup. The inputs $x_1, \ldots, x_n$ with $n = 80$ are sampled from $\mathcal{N}(0, \boldsymbol{I}_d)$ where $\boldsymbol{I}_d$ is an identity matrix with $d = 200$, and the outputs are generated as $y_i = \langle \beta_*, x_i \rangle$ where $\beta_* \in \mathbb{R}^d$ is $r = 20$ sparse. We

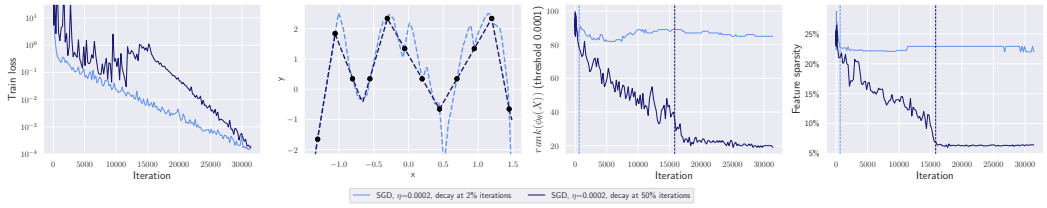

Figure 4: Two-layer ReLU networks for 1D regression. We observe loss stabilization, simplification of the model trained with a longer schedule and sparser features.

consider four different SGD runs (started from $u_i = 0.1$, $v_i = 0$ for each i): one with a small step size and three other with initial large step size decayed after 10%, 30%, 50% iterations, respectively.

Observations. We show the results in Fig. 2 and note that **(O1)–(O3)** hold even in this simple model trained with vanilla SGD without any explicit regularization or layer normalization schemes. We observe that the training loss stabilizes around $10^{-1.5}$, the test loss improves for longer schedules, both $rank(\phi_\theta(X))$ and $\|u \odot v\|_0$ decrease during the loss stabilization phase leading to a sparse final predictor. While the training loss has seemingly converged to $10^{-1.5}$, a hidden dynamics suggested by Eq.(8) occurs which slowly drifts the iterates to a sparse solution. This implicit sparsification explains the dependence of the final test loss on the time when the large step size is decayed, similarly to what has been observed for deep learning in Fig. 1. Interestingly, we also note that SGD with large step size schedules encounters saddle points after we decay the step size (see the training loss curves in Fig. 2) which resembles the saddle-to-saddle regime described in Jacot et al. (2021) which does not occur in large-initialization lazy regime.

SGD and GD have different implicit biases. Since we observe from Fig. 2 that for loss stabilization, stochasticity alone does not suffice and large step sizes are necessary, one may wonder if conversely only large step sizes can be sufficient to have a sparsifying effect. Even if special instances can be found for which large step sizes are sufficient (Nacson et al., 2022), we answer this negatively showing that gradient descent in general does not go to the sparsest solution as demonstrated in Fig. 8 in the Appendix. Moreover, in Fig. 3, we visualize the difference in trajectory between the two methods taken with large step sizes over a 2D subspace spanned by $w^\star - w_{init}$ and $w_{flow} - w_{init}$, where $w^\star$ is the ground truth, w_{flow} is the result of gradient flow, and w_{init} is the initialization. This example provides an important intuition that loss stabilization alone is not sufficient for sparsification and that the noise-driven drift that we have described earlier is important.

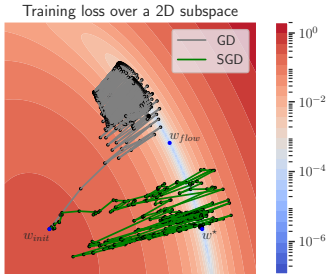

Figure 3: GD and SGD take different trajectories.

3.2 Sparse Feature Learning in Simple ReLU Networks

Two-layer ReLU network in 1D. We consider the one-dimensional regression task from Blanc et al. (2020) with 12 points, where label noise SGD has been shown to learn a simple model. We show that similar results can be achieved with large-step-size SGD via loss stabilization. We train a ReLU network with 100 neurons with SGD with a linear warmup (otherwise, we were unable to achieve approximate loss stabilization) and then step size decay where the length of this schedule is 2% vs. 50%. The results shown in Fig. 4 confirm that **(O1)–(O3)** hold: the training loss stabilizes around $10^{-0.5}$, the predictor becomes much simpler and is expected to generalize better, and both $rank(\phi_\theta(X))$ and the feature sparsity coefficient substantially decrease during the loss stabilization phase. For this one-dimensional task, we can directly observe the final predictor. It is sparse in terms of the number of distinct ReLU kinks as captured by the feature sparsity coefficient and the rank of the noise feature matrix. Interestingly, we also observed *overregularization* for even larger step sizes when we cannot fit all the training points (see Fig. 9 in Appendix). This phenomenon clearly illustrates how the capacity control is induced by the optimization algorithm: the function class over which we optimize depends on the step size schedule. Additionally, Fig. 10 in Appendix shows the evolution of the predictor over iterations. The general picture is confirmed: first the model is simplified during the loss stabilization phase and then fits the training data.

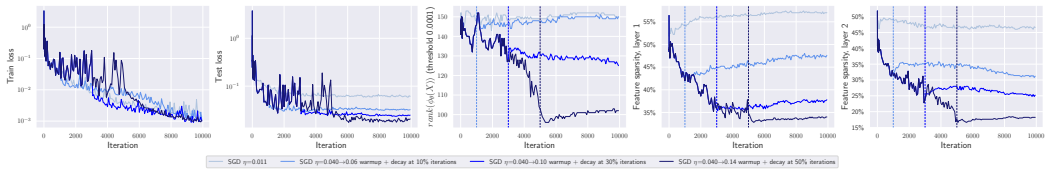

Figure 6: Three-layer ReLU networks in a teacher-student setup. Loss stabilization occurs and leads to a lower rank of the noise feature matrix and lower feature sparsity on both layers.

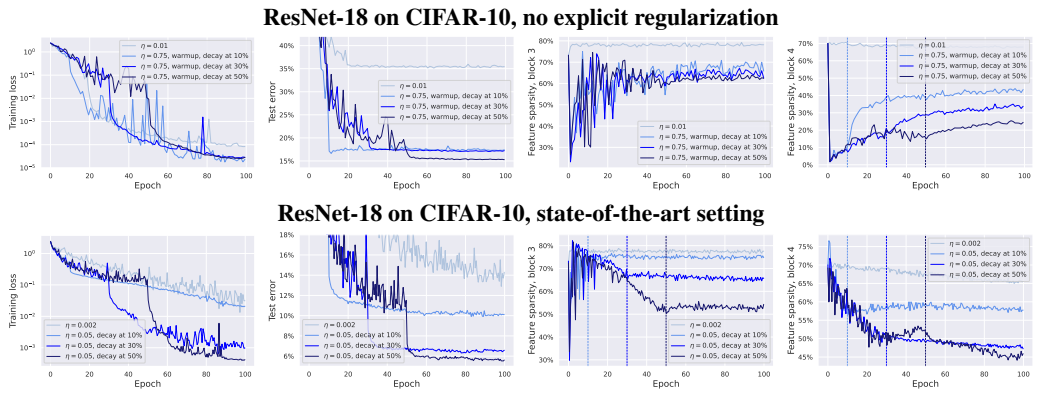

Figure 7: ResNet-18 trained on CIFAR-10. Both *without explicit regularization* and in *the state-of-the-art setting*, the training loss stabilizes, the test loss significantly depends on the length of the schedule, and feature sparsity is minimized over iterations.

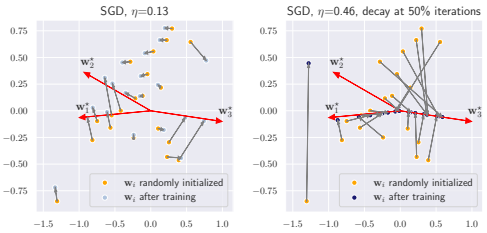

Figure 5: Only for a large step size, the neurons w_i cluster along the teacher neurons $w_i^\star$ leading to a model that uses a sparse set of features.

Illustration of neuron dynamics. We illustrate the change of neurons during training of two-layer ReLU networks in the teacher-student setup of Chizat et al. (2019) (see Fig. 1 therein) using a large initialization scale for which small step sizes of GD or SGD lead to lazy training. We postpone the illustration of **O1**–**O3** to Fig. 11 in Appendix and instead focus on Fig. 5 showing neuron dynamics. We see that for SGD with a small step size, the neurons w_i stay close to their initialization, while for a large step size, there is a clear clustering of directions w_i along the teacher directions $w_i^\star$. The overall picture is very similar to Fig. 1 of Chizat et al. (2019) where the same feature learning effect is achieved via gradient flow from a small initialization which is much more computationally expensive due to the saddle point at zero.

Deeper ReLU networks. We use a teacher-student setup with a random *three-layer* teacher ReLU network having 2 neurons on each hidden layer. The student network is overparametrized with 10 neurons on each layer. We train models using SGD with a medium constant step size and a large step size with warmup which is decayed after 10%, 30%, 50% iterations, respectively. The results shown in Fig. 6 confirm that **(O1)**–**(O3)** hold: the training loss stabilizes around $10^{-1.5}$, the test loss is smaller for longer schedules, and both $rank(\phi_\theta(X))$ and the feature sparsity coefficient substantially decrease during the loss stabilization phase. All methods achieve the same value of the training loss (10^{-3}). We see that the feature sparsity coefficient decreases *on each layer* which makes this metric a promising one to consider for deeper networks.

3.3 Sparse feature learning in deep ReLU networks

Setup. We consider here an image classification task and train a ResNet-18 and ResNet-34 on CIFAR-10 and CIFAR-100 using SGD with batch size 256 and different step size schedules. We use an exponentially increasing warmup schedule with exponent 1.05 to stabilize the training loss.

We cannot measure the rank of $\phi(X)$ here since this matrix is too large ($50\,000 \times 20\,000\,000$) so we measure only the feature sparsity coefficient taken at two layers: at the end of super-block 3 (i.e., in the middle of the network) and super-block 4 (i.e., right before global average pooling) of ResNets. We test two settings: a basic setting without explicit regularizers such as weight decay and a state-of-the-art setting with weight decay, momentum, and standard augmentations.

Observations. The results illustrated in Fig. 7 confirm that our main findings still holds on classification tasks: the training loss stabilizes either slightly below 10^{-1} or above 10^{-1}, the test error is becoming progressively better for longer schedules, as well as the feature sparsity coefficient. Small step size leads to bad generalization, especially without explicit regularization: 35% vs. 15%. This poor performance confirms that it is crucial to leverage the implicit bias of large step sizes. The difference in feature sparsity coefficient is also substantial with the final solution having 24% instead of 70% at block 4 without explicit regularization. The observations are similar for the state-of-the-art setting as well where even with explicit regularization, we still see a difference in generalization and feature sparsity depending on the step size and schedule. We further note that feature sparsity coefficient is gradually minimized over iterations in that case while without explicit regularization we observe a very quick drop down to almost zero at the very first epoch. Overall, findings on CIFAR-100 (see Fig. 13 in Appendix) are similar. Additionally, Fig. 12 illustrates that for small step sizes, the early and middle layers stay very close to randomly initialization which indicates the absence of feature learning akin to the neuron movement plot in Fig. 5.

4 Insights from our understanding of the training dynamics

Here we provide an extended discussion on the implications of our theoretical and empirical findings.

The multiple stages of the SGD training dynamics. As analyzed and shown empirically, deep learning training dynamics can be split onto three distinct phases: (i) an initial phase of reducing the loss down to some level where stabilization can occur, (ii) a loss stabilization phase where noise and gradient directions combine to find architecture-dependent sparse representations of the data, (iii) a final phase when the step size is decreased to fit the training data. This typology allows to clearly disentangle the effect of the exploration phase of SGD (ii) which relies on the implicit bias of SGD to simplify the model. However, some training schedules deviate from this categorization. For example, phases (ii) and (iii) can be repeated up multiple times until final convergence (He et al., 2016). Finally, in some training schedules, (ii) might not explicitly occur and the effect of loss stabilization (ii) and data fitting (iii) can occur simultaneously (Nakkiran et al., 2019).

From lazy training to feature learning. Similar sparse implicit biases have been shown for regression with infinitely small initialization (Boursier et al., 2022) and for classification (Chizat and Bach, 2020; Lyu and Li, 2020). However, both approaches are not practical from the computational point of view since (i) the origin is a saddle point for regression leading to the vanishing gradient problem, and (ii) max-margin biase for classification is only expected to happen in the asymptotic phase (Moroshko et al., 2020). On the contrary, large step sizes enable to initialize far from the origin, while allowing to *efficiently* transition from a regime close to the lazy NTK regime (Jacot et al., 2018) to the rich feature learning regime where the model learns sparse representations.

Common patterns in the existing techniques. Tuning the step size to obtain loss stabilization can be difficult. To prevent early divergence caused by overlarge step size, we sometimes had to rely on an increasing step size schedule (known as *warmup*). Interpreting such schedule as a tool to favor implicit regularization provides a new explanation to their success and popularity. It is worth noting that, beyond carrying their own implicit or explicit regularization properties, normalization schemes like *batch normalization* or *weight decay*, can be analyzed from a similar lens: they allow to use larger step sizes that boost further the implicit bias effect of SGD while preventing divergence (Bjorck et al., 2018; Zhang et al., 2018). On a different note, we derived our analysis with batch size equal to one for the sake of clarity, but an arbitrary batch size B would simply be equivalent to replacing $\gamma \leftarrow \gamma/B$. Similarly to the consequence of large step size, preferring *smaller batch sizes* (Keskar et al., 2016) while avoiding divergence seem key to benefit from the implicit bias of SGD. Finally, the effect of large step sizes or small batches are often connected to measures of *flatness* of the loss surface via stability analysis (Wu et al., 2018) and some methods like SAM explicitly optimize for it (Foret et al., 2021). Such methods resemble the implicit bias of SGD implied by the label noise equation (Eq.(8)).

REFERENCES

N. Agarwal, S. Goel, and C. Zhang. Acceleration via fractal learning rate schedules. In International Conference on Machine Learning, pages 87–99. PMLR, 2021.

B. Barak, B. L. Edelman, S. Goel, S. Kakade, E. Malach, and C. Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. arXiv preprint arXiv:2207.08799, 2022.

G. Beugnot, J. Mairal, and A. Rudi. On the benefits of large learning rates for kernel methods. arXiv preprint arXiv:2202.13733, 2022.

N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger. Understanding batch normalization. Advances in neural information processing systems, 31, 2018.

G. Blanc, N. Gupta, G. Valiant, and P. Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In Conference on learning theory, pages 483–513. PMLR, 2020.

E. Boursier, L. Pillaud-Vivien, and N. Flammarion. Gradient flow dynamics of shallow relu networks for square loss and orthogonal inputs. arXiv preprint arXiv:2206.00939, 2022.

L. Chen and J. Bruna. On gradient descent convergence beyond the edge of stability. arXiv preprint arXiv:2206.04172, 2022.

L. Chizat and F. Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In Conference on Learning Theory, pages 1305–1338. PMLR, 2020.

L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. Advances in Neural Information Processing Systems, 32, 2019.

J. M. Cohen, S. Kaur, Y. Li, J. Z. Kolter, and A. Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. arXiv preprint arXiv:2103.00065, 2021.

A. Damian, T. Ma, and J. D. Lee. Label noise sgd provably prefers flat global minimizers. Advances in Neural Information Processing Systems, 34:27449–27461, 2021.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp minima can generalize for deep nets. In International Conference on Machine Learning, pages 1019–1028. PMLR, 2017.

P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

J. Geiping, M. Goldblum, P. E. Pope, M. Moeller, and T. Goldstein. Stochastic training is not necessary for generalization. arXiv preprint arXiv:2109.14119, 2021.

J. Z. HaoChen, C. Wei, J. D. Lee, and T. Ma. Shape matters: Understanding the implicit bias of the noise covariance. arXiv preprint arXiv:2006.08680, 2020.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems, 31, 2018.

A. Jacot, F. Ged, B. Şimşek, C. Hongler, and F. Gabriel. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. arXiv preprint arXiv:2106.15933, 2021.

Z. Ji and M. Telgarsky. Directional convergence and alignment in deep learning. Advances in Neural Information Processing Systems, 33:17176–17186, 2020.

S. Kaur, J. Cohen, and Z. C. Lipton. On the maximum Hessian eigenvalue and generalization. arXiv preprint arXiv:2206.10654, 2022.

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836, 2016.

A. Lewkowycz, Y. Bahri, E. Dyer, J. Sohl-Dickstein, and G. Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. arXiv preprint arXiv:2003.02218, 2020.

Q. Li, C. Tai, and E. Weinan. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. The Journal of Machine Learning Research, 20(1): 1474–1520, 2019a.

Y. Li, C. Wei, and T. Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In NeurIPS, 2019b.

Z. Li, T. Wang, and S. Arora. What happens after sgd reaches zero loss?–a mathematical framework. arXiv preprint arXiv:2110.06914, 2021.

K. Lyu and J. Li. Gradient descent maximizes the margin of homogeneous neural networks. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=SJeLIgBKPS.

C. Ma and L. Ying. The Sobolev regularization effect of stochastic gradient descent. arXiv preprint arXiv:2105.13462, 2021.

C. Ma, L. Wu, and L. Ying. The multiscale structure of neural network loss functions: The effect on optimization and origin. arXiv preprint arXiv:2204.11326, 2022.

E. Moroshko, B. E. Woodworth, S. Gunasekar, J. D. Lee, N. Srebro, and D. Soudry. Implicit bias in deep linear classification: Initialization scale vs training accuracy. In Advances in Neural Information Processing Systems, volume 33, pages 22182–22193, 2020.

R. Mulayoff, T. Michaeli, and D. Soudry. The implicit bias of minima stability: A view from function space. Advances in Neural Information Processing Systems, 34:17749–17761, 2021.

M. S. Nacson, K. Ravichandran, N. Srebro, and D. Soudry. Implicit bias of the step size in linear diagonal neural networks. In International Conference on Machine Learning, pages 16270–16295. PMLR, 2022.

P. Nakkiran. Learning rate annealing can provably help generalization, even for convex problems. arXiv preprint arXiv:2005.07360, 2020.

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

S. Pesme, L. Pillaud-Vivien, and N. Flammarion. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. In Advances in Neural Information Processing Systems, 2021.

L. Pillaud-Vivien, J. Reygner, and N. Flammarion. Label noise (stochastic) gradient descent implicitly solves the lasso for quadratic parametrisation. arXiv preprint arXiv:2206.09841, 2022.

S. L. Smith and Q. V. Le. A Bayesian perspective on generalization and stochastic gradient descent. arXiv preprint arXiv:1710.06451, 2017.

D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. The Journal of Machine Learning Research, 19(1):2822–2878, 2018.

T. Vaskevicius, V. Kanade, and P. Rebeschini. Implicit regularization for optimal sparse recovery. Advances in Neural Information Processing Systems, 32, 2019.

S. Wojtowytsch. Stochastic gradient descent with noise of machine learning type. part ii: Continuous time analysis. arXiv preprint arXiv:2106.02588, 2021a.

S. Wojtowytsch. Stochastic gradient descent with noise of machine learning type. part i: Discrete time analysis. arXiv preprint arXiv:2105.01650, 2021b.

B. Woodworth, S. Gunasekar, J. D. Lee, E. Moroshko, P. Savarese, I. Golan, D. Soudry, and N. Srebro. Kernel and rich regimes in overparametrized models. In Proceedings of Thirty Third Conference on Learning Theory, volume 125 of Proceedings of Machine Learning Research, pages 3635–3673. PMLR, 09–12 Jul 2020.

J. Wu, D. Zou, V. Braverman, and Q. Gu. Direction matters: On the implicit bias of stochastic gradient descent with moderate learning rate. ICLR, 2021.

L. Wu, C. Ma, et al. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. Advances in Neural Information Processing Systems, 31, 2018.

C. Xing, D. Arpit, C. Tsirigotis, and Y. Bengio. A walk with sgd. arXiv preprint arXiv:1802.08770, 2018.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In European conference on computer vision, pages 818–833. Springer, 2014.

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

G. Zhang, C. Wang, B. Xu, and R. Grosse. Three mechanisms of weight decay regularization. arXiv preprint arXiv:1810.12281, 2018.

A SGD AND LABEL NOISE GD

Proposition 5. *Let $(\theta_t)_{t\geq 0}$ follows the SGD dynamics Eq.(2) with sampling function $(i_t)_{t\geq 0}$. Define for each $t \geq 0$, the random vector $\xi_t \in \mathbb{R}^n$ such that for all $i \in [\![1, n]\!]$,*

$$[\xi_t]_i := (h_{\theta_t}(x_i) - y_i)(1 - n\mathbf{1}_{i=i_t}). \tag{9}$$

Then, $(\theta_t)_{t\geq 0}$ follows the dynamics of the full batch gradient dynamics on $\mathcal{L}$ where at each iteration $t \geq 0$ we inject noise ξ_t to the label vector $y \in \mathbb{R}^n$: that is at step $t \geq 0$, we compute the gradient of $\mathcal{L}$ considering the random labels $y_t := y + \xi_t$.

Proof. Note that

$$\nabla \mathcal{L}(\theta_t) = \frac{1}{n}\sum_{i=1}^{n}(h_{\theta_t}(x_i) - y_i)\nabla_\theta h_{\theta_t}(x_i).$$

Computing the gradient considering the random labels $y_t := y + \xi_t$,

$$\nabla \widehat{\mathcal{L}}(\theta_t) = \frac{1}{n}\sum_{i=1}^{n}(h_{\theta_t}(x_i) - y_i - [\xi_t]_i)\nabla_\theta h_{\theta_t}(x_i).$$

Using $[\xi_t]_i := (h_{\theta_t}(x_i) - y_i)(1 - n\mathbf{1}_{i=i_t})$,

$$\begin{aligned}\nabla \widehat{\mathcal{L}}(\theta_t) &= \frac{1}{n}\sum_{i=1}^{n}(h_{\theta_t}(x_i) - y_i - (h_{\theta_t}(x_i) - y_i)(1 - n\mathbf{1}_{i=i_t}))\nabla_\theta h_{\theta_t}(x_i)\\ &= \sum_{i=1}^{n}\mathbf{1}_{i=i_t}(h_{\theta_t}(x_i) - y_i)\nabla_\theta h_{\theta_t}(x_i) = (h_{\theta_t}(x_{i_t}) - y_{i_t})\nabla_\theta h_{\theta_t}(x_{i_t})\end{aligned}$$

which is exactly the stochastic gradient wrt to sample (x_{i_t}, y_{i_t}).

□

Lemma 6. *For $t \geq 0$, let $\xi_t \in \mathbb{R}^n$ be defined as in Proposition 1, then ξ_t is a mean zero random vector with variance such that $\frac{1}{(n-1)^2}\mathbb{E}\left\|\xi_t\right\|^2 = 2\mathcal{L}(\theta_t)$.*

Proof. Recall that,

$$[\xi_t]_i = (h_{\theta_t}(x_i) - y_i)(1 - n\mathbf{1}_{i=i_t}).$$

where $i_t \sim \mathcal{U}([\![1, n]\!])$. Now taking the expectation,

$$\mathbb{E}\left[[\xi_t]_i\right] = \mathbb{E}\left[(h_{\theta_t}(x_i) - y_i)(1 - n\mathbf{1}_{i=i_t})\right] = (h_{\theta_t}(x_i) - y_i)(1 - n\mathbb{E}\left[\mathbf{1}_{i=i_t}\right]).$$

Using the fact that $\mathbb{E}\left[\mathbf{1}_{i=i_t}\right] = 1/n$, we can show that ξ_t is zero mean. Coming to the variance,

$$\begin{aligned}\mathbb{E}\left\|\xi_t\right\|^2 = \mathbb{E}\left[\sum_{i=1}^{n}[\xi_t]_i^2\right] &= \sum_{i=1}^{n}\mathbb{E}[\xi_t]_i^2\\ &= \sum_{i=1}^{n}(h_{\theta_t}(x_i) - y_i)^2\mathbb{E}\left[(1 - n\mathbf{1}_{i=i_t})^2\right]\\ &= \sum_{i=1}^{n}(h_{\theta_t}(x_i) - y_i)^2\mathbb{E}\left[(1 - 2n\mathbf{1}_{i=i_t} + n^2\mathbf{1}_{i=i_t})\right]\\ &= \sum_{i=1}^{n}(h_{\theta_t}(x_i) - y_i)^2(1 - 2 + n)\\ &= (n-1)\sum_{i=1}^{n}(h_{\theta_t}(x_i) - y_i)^2 = 2n(n-1)\mathcal{L}(\theta_t).\end{aligned}$$

□

B QUADRATIC PARAMETERIZATION IN ONE DIMENSION

Proposition 7. *Assume* $\exists\, x_{\min}, x_{\max} > 0$ *such that supp*$(\hat{\rho}) \subset [x_{\min}, x_{\max}]$. *Then for any* $\eta \in ((\theta_* x_{\min})^{-2}, 1.25(\theta_* x_{\max})^{-2})$, *any initialization in* $\theta_0 \in (0, \theta_*)$, *for* $t \in \mathbb{N}$, *we have almost surely*

$$F(\theta_t) \in \left(\epsilon_{\mathrm{o}}^2\, \theta_*^2, 0.17\, \theta_*^2\right). \tag{10}$$

where $\epsilon_{\mathrm{o}} = \min\left\{(\eta(\theta_* x_{\min})^2 - 1)/3, 0.02\right\}$. *Also, almost surely, there exists* $t, k > 0$ *such that* $\theta_{t+2k} \in (0.65\, \theta_*, (1-\epsilon_{\mathrm{o}})\, \theta_*)$ *and* $\theta_{t+2k+1} \in ((1+\epsilon_{\mathrm{o}})\, \theta_*, 1.162\, \theta_*)$.

Proof. Consider SGD recursion Eq.(4) and note that $y = \langle x, \theta_*^2 \rangle$.

$$\theta_{t+1} = \theta_t + \eta\, \theta_t\, x(\langle x, \theta_*^2 \rangle - \langle x, \theta_t^2 \rangle)$$
$$\theta_{t+1} = \theta_t + \eta\, \theta_t\, x^2\, (\theta_*^2 - \theta_t^2)$$

For the clarity of exposition, we consider the rescaled recursion of the original SGD recursion.

$$\theta_{t+1}/\theta_* = \theta_t/\theta_* + \eta\, \theta_*^2\, x^2\, \theta_t/\theta_* \left(1 - (\theta_t/\theta_*)^2\right)$$

Hence, we focus on the stochastic recursion,

$$\theta_{t+1} = \theta_t + \gamma\theta_t(1-\theta_t^2) \quad \gamma \sim \hat{\rho}_\gamma \tag{11}$$

It is important to note that γ is not constant and is governed by a distribution $\hat{\rho}_\gamma$ which depends on $\hat{\rho}$. Let Γ denote the support of the distribution of γ. From the range of η, it can be verified that $\Gamma \subseteq (1, 1.25)$. Now the proof of the theorem follows from 9 □

Lemma 8 (Bounded Region)**.** *Consider the recursion Eq.(11), for* $\Gamma \subseteq (1, 1.25)$ *and* $0 < \theta_0 < 1$, *then for all* $t > 0$, $\theta_t \in (0, 1.162)$.

Proof. Consider a single step of Eq.(11), for some $\gamma \in (1, 1.25)$,

$$\theta_+ = \theta + \gamma\theta(1-\theta^2)$$

The aim is to show that θ_+ stays in the interval $(0, 1.162)$. In order to show this, we do a casewise analysis.

For $\theta \in (0, 1]$: Since $0 < \theta \leq 1$, we have $\theta_+ \geq \theta > 0$. To prove the bound above, consider the following quantity,

$$\theta_{max} = \max_{\gamma \in (1, 1.25)} \max_{\theta \in (0,1]} \theta + \gamma\theta(1-\theta^2) \tag{12}$$

Say $h_\gamma(\theta) = \theta + \gamma\theta(1-\theta^2)$, note that $h'_\gamma(\theta) = 1 + \gamma - 3\gamma\theta^2$ and $h''_\gamma(\theta) = -6\gamma\theta < 0$. Hence, for any γ in our domain, the maximum is attained at $\theta_\gamma = \frac{1}{\sqrt{3}}\sqrt{\frac{1}{\gamma} + 1}$ and $h_\gamma(\theta_\gamma) = \frac{2(1+\gamma)^{3/2}}{3\sqrt{3\gamma}}$.

$$\max_{\gamma \in (1, 1.25)} \max_{\theta \in (0,1]} \theta + \gamma\theta(1-\theta^2) = \max_{\gamma \in (.5, 1.25)} \frac{2(1+\gamma)^{3/2}}{3\sqrt{3\gamma}}$$

It can be verified that $\frac{2(1+\gamma)^{3/2}}{3\sqrt{3\gamma}}$ is increasing with gamma in the interval $(1, 1.25)$. Hence,

$$\max_{\gamma \in (1, 1.25)} \frac{2(1+\gamma)^{3/2}}{3\sqrt{3\gamma}} \leq \left.\frac{2(1+\gamma)^{3/2}}{3\sqrt{3\gamma}}\right|_{\gamma = 1.25} < 1.162 \tag{13}$$

Combining them, we get,

$$\theta_+ \leq \max_{\gamma \in (0, 1.25)} \max_{\theta \in (0,1]} \theta + \gamma\theta(1-\theta^2) < 1.162 \tag{14}$$

For $\theta \in (1, 1.162)$: Since $\theta > 1$, we have, $\theta_+ < \theta < 1.162$. For lower bound, note that for θ_+ to be less than 0, we need $1 + \gamma - \gamma\theta^2 < 0$. But for $\gamma \in (1, 1.25)$ and $\theta \in (1, 1.162)$,

$$\gamma(\theta^2 - 1) < 1.25((1.162)^2 - 1) < 1.$$

Hence, it never goes below 0. □

Lemma 9. *Consider the recursion Eq.(11) with* $\Gamma \subseteq (1, 1.25)$ *and* θ_0 *initialized uniformly in* $(0, 1)$*. Then, there exists* $\epsilon_0 > 0$*, for all* $\epsilon < \epsilon_0$ *there exists* $t > 0$ *such that for any* $k > 0$*,*

$$\theta_{t+2k} \in (0.65, 1-\epsilon) \quad \textit{and} \quad \theta_{t+2k+1} \in (1+\epsilon, 1.162)$$

almost surely.

Proof. Define $\gamma_{\min} > 1$ as the infimum of the support Γ. Let $\epsilon_o = \min\{{}^{(\gamma_{min}-1)}/_3, 0.02\}$. Note that such an ϵ exists as $\gamma_{\min} > 1$. Now any $\epsilon < \epsilon_o$, we have $\gamma_{\min}(2-\epsilon)(1-\epsilon) > 2$.

Divide the interval (0,1.162) into 4 regions, $\mathrm{I}_0 = (0, 0.65]$, $\mathrm{I}_1 = (0.65, 1-\epsilon)$, $\mathrm{I}_2 = [1-\epsilon, 1)$, $\mathrm{I}_3 = (1, 1.162)$. The strategy of the proof is that the iterates will eventually end up in I_1 and that once it ends up in I_1, it comes back to I_1 in 2 steps.

Let θ_0 be initialized uniformly random in $(0, 1)$. Consider the sequence $(\theta_t)_{t\geq 0}$ generated by

$$\theta_{t+1} = \mathsf{h}_{\gamma_t}(\theta_t) \quad \text{where} \quad \gamma_t \sim \hat{\rho}_\gamma. \tag{15}$$

(P1) There exists $t > 0$ such that the $\theta_t \in \mathrm{I}_1 \cup \mathrm{I}_2 \cup \mathrm{I}_3$.

(P2) Let $\theta_t \in \mathrm{I}_3$, then $\theta_{t+1} \in \mathrm{I}_1 \cup \mathrm{I}_2$.

(P3) Let $\theta_t \in \mathrm{I}_2$, there exists $k > 0$ such that for $k' < k$, $\theta_{t+2k'} \in \mathrm{I}_2$ and $\theta_{t+2k} \in \mathrm{I}_1$.

(P4) When $\theta_t \in \mathrm{I}_1$, then for all $k \geq 0$, $\theta_{t+2k} \in \mathrm{I}_1$ and $\theta_{t+2k+1} \in (1+\epsilon, 1.162)$.

Proof: First we exclude the 0-measure pathological case where θ_t falls directly on 1. For any other sequence, from the above four properties, we can conclude that the lemma holds.

Proof of P1: Assume that $\theta_0 \in \mathrm{I}_0$, otherwise its already in $\mathrm{I}_1 \cup \mathrm{I}_2 \cup \mathrm{I}_3$. Let $\theta_t \in \mathrm{I}_0$, we have

$$\theta_{t+1} = \theta_t + \gamma\theta_t(1-\theta_t^2) \quad \gamma \in \Gamma,$$
$$\theta_{t+1} = \theta_t(1+\gamma(1-\theta_t^2)) \geq \theta_t(2-\theta_t^2) > 1.5\,\theta_t$$

Note that $\theta_t, \theta_{t+1}, \ldots$ are exponentially increasing with base > 1. Hence, the sequence eventually exits I_0. We know that it will stay bounded from Lemma 8, hence it will end up in $\mathrm{I}_1 \cup \mathrm{I}_2 \cup \mathrm{I}_3$.

Proof of P2: For any $\theta_t \in (1, 1.162)$, $1 < \gamma < 1.25$,

Since $\mathsf{h}_\gamma(.)$ is decreasing in (1,1.162), we have $\mathsf{h}_\gamma(1.162) < \mathsf{h}_\gamma(\theta_t) < \mathsf{h}_\gamma(1)$.

Also $\mathsf{h}_\gamma(\theta)$ is linear in gamma with negative coefficient for $\theta > 1$. Hence it decreases as γ increases. Using this,

$$.652 = \mathsf{h}_{1.25}(1.162) < \mathsf{h}_\gamma(1.162) < \mathsf{h}_\gamma(\theta_t) < \mathsf{h}_\gamma(1) = 1$$

Hence, $\theta_{t+1} \in \mathrm{I}_1 \cup \mathrm{I}_2$.

Proof of P3: The proof of this follows from Lemma 11.

Proof of P4: The proof of this follows from Lemma 14. □

Lemma 10. *For any* $\theta \in \mathrm{I}_1 \cup \mathrm{I}_2$ *and any* $a, b \in \Gamma$, $\mathsf{h}_a(\mathsf{h}_b(\theta)) \in \mathrm{I}_1 \cup \mathrm{I}_2$,

$$\mathsf{h}_{\gamma_{\max}}(\mathsf{h}_{\gamma_{\max}}(\theta)) \leq \mathsf{h}_a(\mathsf{h}_b(\theta)) \leq \mathsf{h}_{\gamma_{\min}}(\mathsf{h}_{\gamma_{\min}}(\theta)). \tag{16}$$

Proof. For any $\gamma \in \Gamma$, recall

$$\mathsf{h}_\gamma(\theta) = \theta + \gamma\theta(1-\theta^2) = 1 + (1-\theta)(\gamma\theta(1+\theta) - 1)$$

Note that for $\theta \in \mathrm{I}_1 \cup \mathrm{I}_2$, $\theta(1+\theta) > 1$, Hence $\gamma\theta(1+\theta) > 1$. This gives us that $\mathsf{h}_\gamma(\theta) > 1$. Now we will track where $\theta \in \mathrm{I}_1 \cup \mathrm{I}_2$ can end up after two stochastic gradient steps.

- For any $b \in \Gamma$, as $\theta \in \mathrm{I}_1 \cup \mathrm{I}_2$, we have

  $$\mathsf{h}_{\gamma_{\max}}(\theta) \geq \mathsf{h}_b(\theta) \geq \mathsf{h}_{\gamma_{\min}}(\theta) > 1,$$

  note $\mathsf{h}_{\gamma_{\max}}(\theta) \geq \mathsf{h}_b(\theta) \geq \mathsf{h}_{\gamma_{\min}}(\theta)$ holds since $\theta < 1$.

- Now for any $a \in \Gamma$ and $x > 1$, $\mathsf{h}_a(x)$ is a decreasing function in x. Hence

$$\mathsf{h}_a(\mathsf{h}_{\gamma_{\max}}(\theta)) \leq \mathsf{h}_a(\mathsf{h}_b(\theta)) \leq \mathsf{h}_a(\mathsf{h}_{\gamma_{\min}}(\theta)).$$

  Using $\gamma_{\min} \leq a$, $\mathsf{h}_a(\mathsf{h}_{\gamma_{\min}}(\theta)) \leq \mathsf{h}_{\gamma_{\min}}(\mathsf{h}_{\gamma_{\min}}(\theta))$, Similarly using $\gamma_{\max} > a$, we have, $\mathsf{h}_{\gamma_{\max}}(\mathsf{h}_{\gamma_{\max}}(\theta)) \leq \mathsf{h}_a(\mathsf{h}_{\gamma_{\max}}(\theta))$. Combining them we get,

$$\mathsf{h}_{\gamma_{\max}}(\mathsf{h}_{\gamma_{\max}}(\theta)) \leq \mathsf{h}_a(\mathsf{h}_b(\theta)) \leq \mathsf{h}_{\gamma_{\min}}(\mathsf{h}_{\gamma_{\min}}(\theta)). \tag{17}$$

Similar argument can extend it to,

$$\mathsf{h}_{1.25}(\mathsf{h}_{1.25}(\theta)) < \mathsf{h}_a(\mathsf{h}_b(\theta)) < \mathsf{h}_1(\mathsf{h}_1(\theta)). \tag{18}$$

□

Lemma 11. *Let $\theta_t \in \mathrm{I}_2$, there exists $k > 0$ such that $\theta_{t+2k} \in \mathrm{I}_1$.*

Proof. For any $\gamma \in \Gamma$, let $\theta_+ = \mathsf{h}_\gamma(\theta)$.

$$\begin{aligned}
\mathsf{h}_\gamma(\mathsf{h}_\gamma(\theta)) - \theta &= \mathsf{h}_\gamma(\theta_+) - \theta, \\
&= \theta_+ + \gamma\theta_+(1-\theta_+^2) - \theta, \\
&= \gamma\theta(1-\theta^2) + \gamma\theta_+(1-\theta_+^2), \\
\frac{1}{\gamma}\left[\mathsf{h}_\gamma(\mathsf{h}_\gamma(\theta)) - \theta\right] &= \theta(1-\theta^2) + \theta_+(1-\theta_+^2).
\end{aligned}$$

$$\begin{aligned}
\theta_+ &= \theta + \gamma\theta(1-\theta^2) = \theta(1+\gamma(1-\theta^2)), \\
1+\theta_+ &= 1+\theta+\gamma\theta(1-\theta^2) = (1+\theta)(1+\gamma\theta(1-\theta)), \\
1-\theta_+ &= 1-\theta-\gamma\theta(1-\theta^2) = (1-\theta)(1-\gamma\theta(1+\theta)).
\end{aligned}$$

Multiplying the above three terms and adding $\theta(1-\theta^2)$, we get,

$$\theta_+(1-\theta_+^2) + \theta(1-\theta^2) = \theta(1-\theta^2)\{1 + \underbrace{\left[(1+\gamma(1-\theta^2))(1+\gamma\theta(1-\theta))(1-\gamma\theta(1+\theta))\right]}_{P(\theta)}\}$$

For $\theta \in \mathrm{I}_2$ and using $\gamma_{\min}(2-\epsilon)(1-\epsilon) > 2$,

$$\begin{aligned}
(1+\gamma(1-\theta^2))(1+\gamma\theta(1-\theta)) &> 1, \\
(1-\gamma\theta(1+\theta)) < 1-\gamma_{min}(2-\epsilon)(1-\epsilon) &< -1, \\
P(\theta) &< -1, \\
1+P(\theta) &< 0.
\end{aligned}$$

Hence,

$$\mathsf{h}_\gamma(\mathsf{h}_\gamma(\theta)) - \theta = \gamma(1-\theta^2)(1+P(\theta)) < 0 \tag{19}$$

Hence for $[1-\epsilon, 1)$, for any $\gamma \in \Gamma$, $\mathsf{h}_\gamma(\mathsf{h}_\gamma(\theta)) < \theta$. Hence for any two stochastic gradient step with $a, b \in \Gamma$, from Eq.(16), $\theta_{t+2} = \mathsf{h}_a(\mathsf{h}_b(\theta_t)) \leq \mathsf{h}_{\gamma_{\min}}(\mathsf{h}_{\gamma_{\min}}(\theta_t)) < \theta_t$. From any point in I_2, we have $|\theta_{t+2} - 1| > |\theta_t - 1|$, for any $a, b \in \Gamma$. Intutively this means that in *two gradient steps* the iterates move *further away from 1* until it eventually leaves the interval I_2 as the sequence $\{\theta_{t+2k}\}_{k\geq 0}$ is strictly decreasing with no limit point in I_2. From Lemma 13 , we know that in two steps the iterates will never leave $\mathrm{I}_1 \cup \mathrm{I}_2$. Hence they will eventually end up in I_1 leaving I_2. □

Property 12. *The followings properties hold for $\theta \in \mathrm{I}_1 \cup \mathrm{I}_2$, $\gamma \in \Gamma$ and θ_γ the root of $\mathsf{h}_\gamma'(\theta)$. Also define $\mathsf{g}_\gamma(\theta) := \mathsf{h}_\gamma(\mathsf{h}_\gamma(\theta))$ for the sake of brevity.*

Q1 $\mathsf{g}_\gamma(\theta) \geq \mathsf{g}_\gamma(\theta_\gamma)$.

Q2 *The function $\mathsf{g}_\gamma(.)$ is decreasing in $[0.65, \theta_\gamma)$ and increasing in $(\theta_\gamma, 1]$.*

Proof. Note $\mathsf{h}_\gamma^{'}(\theta) = 1 + \gamma - \gamma 3\theta^2$, has at most one root in (0,1), i.e., $\mathsf{h}_\gamma^{'}(\theta_\gamma) = 0$. Note that for all $\gamma \in \Gamma$, $\theta_\gamma \in \mathrm{I}_1 \cup \mathrm{I}_2$. For any γ, $\mathsf{g}_\gamma^{'}(\theta) = \mathsf{h}_\gamma^{'}(\mathsf{h}_\gamma(\theta))\mathsf{h}_\gamma^{'}(\theta)$. For any $\theta \in \mathrm{I}_1 \cup \mathrm{I}_2$, we have, $\mathsf{h}_\gamma(\theta) > 1 \implies \mathsf{h}_\gamma^{'}(\mathsf{h}_\gamma(\theta)) < 0$. Therefore, $\mathsf{g}_\gamma^{'}(\theta)$ has only one root in $\mathrm{I}_1 \cup \mathrm{I}_2$. Since $\theta_\gamma \in \mathrm{I}_1 \cup \mathrm{I}_2$, note $\mathsf{g}_\gamma^{''}(\theta_\gamma) = \mathsf{h}_\gamma^{'}(\mathsf{h}_\gamma(\theta_\gamma))\mathsf{h}_\gamma^{''}(\theta_\gamma) > 0$. Therefore, $\mathsf{g}_\gamma(.)$ attains the minimum at θ_γ. Hence, we have the following properties. □

Lemma 13. *For any $\theta \in \mathrm{I}_1 \cup \mathrm{I}_2$ and any $a, b \in \Gamma$, $\mathsf{h}_a(\mathsf{h}_b(\theta)) \in \mathrm{I}_1 \cup \mathrm{I}_2$.*

Proof. **Lower Bound:** From Eq.(18), we know

$$\mathsf{h}_{1.25}(\mathsf{h}_{1.25}(\theta)) < \mathsf{h}_a(\mathsf{h}_b(\theta))$$

We know that from property **Q1** that $\mathsf{g}_\gamma(\theta) \geq \mathsf{g}_\gamma(\theta_\gamma)$. Hence

$$\mathsf{g}_{1.25}(\theta_{1.25}) < \mathsf{g}_{1.25}(\theta) < \mathsf{h}_a(\mathsf{h}_b(\theta))$$

It can be quickly checked that $.65 < \mathsf{g}_{1.25}(\theta_{1.25})$. Hence the lower bound holds.

Upper Bound: From Eq.(18), we know

$$\mathsf{h}_a(\mathsf{h}_b(\theta)) < \mathsf{h}_1(\mathsf{h}_1(\theta))$$

We know that from property **Q2** that $\mathsf{g}_1(\theta) \leq \max\{\mathsf{g}_1(1), \mathsf{g}_1(0.65)\}$. It can be easily verified that $\mathsf{g}_1(0.65) < 0.98$. Hence $\mathsf{g}_1(\theta) < 1$.

□

Lemma 14. *For any $\theta \in \mathrm{I}_1$ and any $a, b \in \Gamma$, $\mathsf{h}_a(\mathsf{h}_b(\theta)) \in \mathrm{I}_1$ and $\mathsf{h}_a(\theta) \in (1+\epsilon, 1.162)$.*

Proof. The lower bound in Lemma 13 holds here. For the upper bound, from and Eq.(16),

$$\mathsf{h}_a(\mathsf{h}_b(\theta)) \leq \mathsf{h}_{\gamma_{\min}}(\mathsf{h}_{\gamma_{\min}}(\theta))$$

Using property **Q2**,

$$\mathsf{h}_{\gamma_{\min}}(\mathsf{h}_{\gamma_{\min}}(\theta)) \leq \max\{\mathsf{g}_{\gamma_{\min}}(1-\epsilon), \mathsf{g}_{\gamma_{\min}}(0.65)\}$$

From Eq.(19), $\mathsf{g}_{\gamma_{\min}}(1-\epsilon) < 1-\epsilon$. From Eq.(18), $\mathsf{g}_{\gamma_{\min}}(0.65) < \mathsf{g}_1(0.65) < 0.98 < 1-\epsilon$. In I_1, the function $\mathsf{h}_a(.)$ first increases reaches maximum and decreases. Hence for $\theta \in \mathrm{I}_1$, $\mathsf{h}_a(\theta) \geq \min\{\mathsf{h}_a(0.65), \mathsf{h}_a(1-\epsilon)\}$.

$$\begin{aligned}
\mathsf{h}_a(1-\epsilon) &\geq 1 - \epsilon + a(1-(1-\epsilon)^2)(1-\epsilon), \\
&= 1 - \epsilon + a(2\epsilon - \epsilon^2)(1-\epsilon), \\
&\geq 1 - \epsilon + \gamma_{\min}(2\epsilon - \epsilon^2)(1-\epsilon), \\
&= 1 + \epsilon + \epsilon\left(\gamma_{\min}(2-\epsilon)(1-\epsilon) - 2\right) > 1 + \epsilon. \\
\text{Also } \mathsf{h}_a(0.65) &> \mathsf{h}_1(0.65) > 1.02 > 1 + \epsilon.
\end{aligned}$$

Hence $\mathsf{h}_a(\theta) > 1 + \epsilon$. This completes the proof. □

Lemma 15. *Let $X \in \mathbb{R}^{n \times d}$ be a matrix of full rank. For any vector $u, v \in \mathbb{R}^d$, let $\phi_{u,v}(X) = [X \operatorname{diag}(v), X \operatorname{diag}(u)]$ and if $\operatorname{rank}(\phi_{u,v}) < \min\{n, d\}$ then $u \odot v$ is sparse.*

Proof. We will proceed to prove by contradiction. Assume that $u \odot v$ is not sparse. Hence $[u]_i \neq 0$ for any i, similarly for $[v]_i$. Now, $\phi_{u,v}(X) = \phi_{u,v}(X) = [X \operatorname{diag}(v), X \operatorname{diag}(u)] \in \mathbb{R}^{n \times 2d}$. Since, none of the co-ordinates of u, v are zero we can eliminate the ith column for every $1 \leq i \leq d$ by subtracting it with $({}^{[v]_i}/_{[u]_i})\phi_{u,v}(X)[i+d]$ the $(i+d)^{th}$ column of $\phi_{u,v}(X)$. Hence $\operatorname{rank}(\phi_{u,v}) = \operatorname{rank}(X \operatorname{diag}(u)) < \min\{n, d\}$. Therefore $\phi_{u,v}$ is not of full rank and there exists vectors $p \in \mathbb{R}^n, q \in \mathbb{R}^d$ such that

$$p^\top X \operatorname{diag}(u) = \mathbf{0}_d^\top \quad \text{and} \quad X \operatorname{diag}(u)\, q = \mathbf{0}_n,$$

where $\mathbf{0}_d, \mathbf{0}_n$ are zero vectors of dimension d, n respectively. Since $\operatorname{diag}(u)$ is invertible, $p^\top X = \mathbf{0}_d^\top$, similarly $X\hat{q} = \mathbf{0}_n$ for $\hat{q} = \operatorname{diag}(u)\, q$. This is a contradiction since X is full rank and such vector pair $p, \hat{q}$ should not exist. □

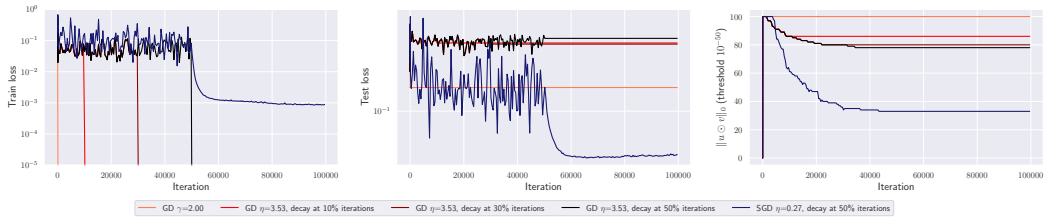

Figure 8: Diagonal linear networks. Loss stabilization also occurs gradient descent but does not lead to a similar level of sparsity as SGD and also does not improve the test loss.

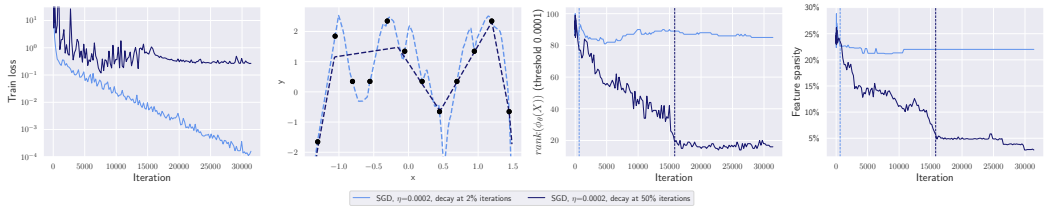

Figure 9: Two-layer ReLU networks for 1D regression. Unlike for Fig. 4, here we use a larger warmup coefficient ($500\times$ vs. $400\times$) which leads to overregularization such that the 50%-schedule run fails to fit all the training points and gets stuck at a too high value of the training loss ($\approx 10^{-0.5}$).

C ADDITIONAL EXPERIMENTAL RESULTS

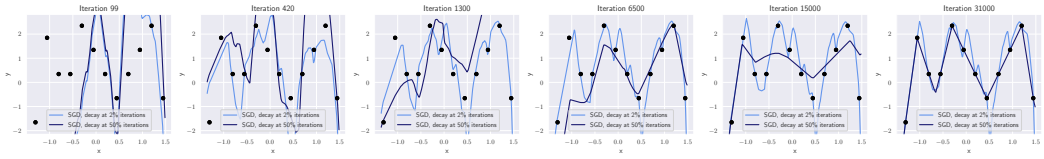

Figure 10: Illustration of the resulting models from Fig. 4 over training iterations.

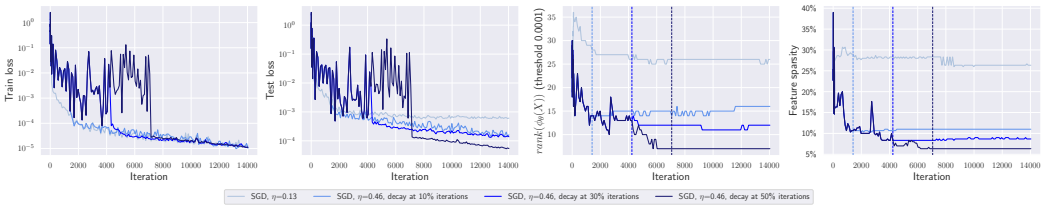

Figure 11: Two-layer ReLU networks in a teacher-student setup. Loss stabilization for two-layer ReLU nets in the teacher-student setup with input dimension $d = 2$. We observe loss stabilization, better test loss for longer schedules and sparser features due to simplification of $\phi(X)$.

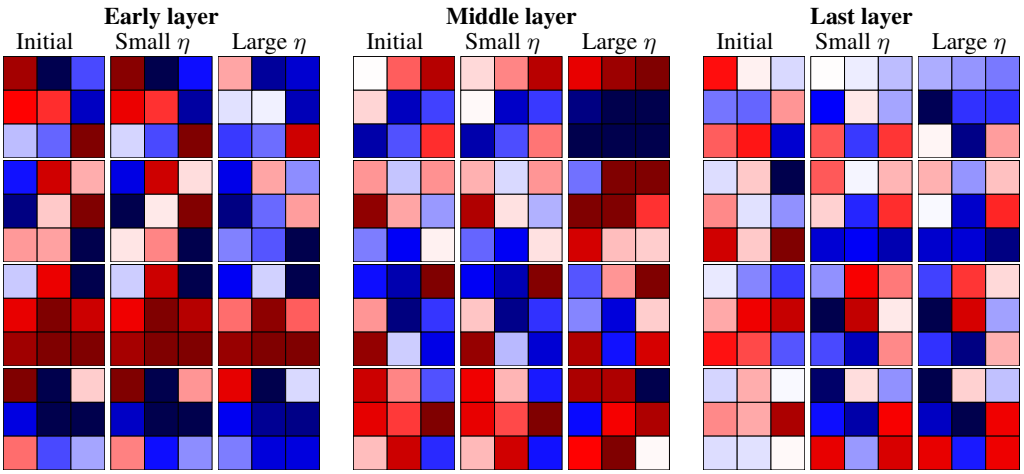

Figure 12: Visualization on two sets of convolutional filters taken from different layers of ResNets-18 trained on CIFAR-10 with small vs. large step size η. For small step sizes, the early and middle layers stay very close to randomly initialized ones which indicates the absence of feature learning.

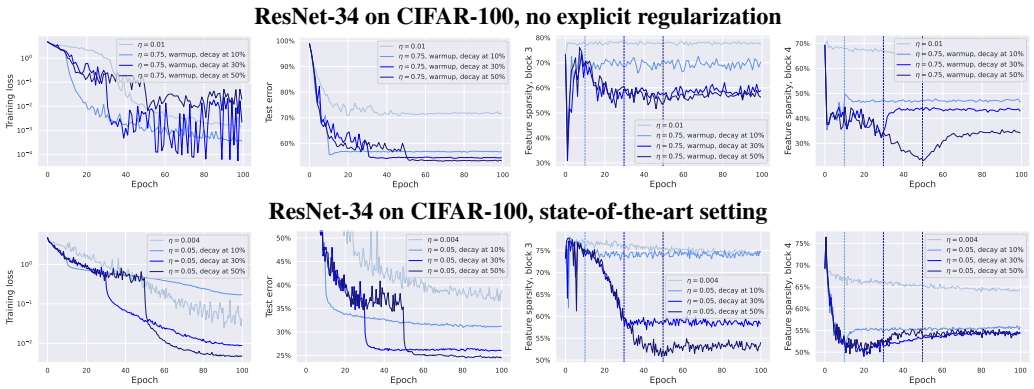

Figure 13: ResNet-34 trained on CIFAR-100. Both *without explicit regularization* and in *the state-of-the-art setting*, the training loss stabilizes, the test loss significantly depends on the length of the schedule, and feature sparsity is minimized over iterations. However, differently from the plots on CIFAR-10, here without explicit regularization we observe oscillating behavior after the step size decay (although at a very low level between 10^{-4} and 10^{-2}).