Generalization in Supervised Learning Through Riemannian Contraction

Anonymous authors
Paper under double-blind review

Abstract

A key property of successful learning algorithms is generalization. In classical supervised learning, generalization can be achieved by ensuring that the empirical error converges to the expected error as the number of training samples goes to infinity. Within this classical setting, we analyze the generalization properties of iterative optimizers such as stochastic gradient descent and natural gradient flow through the lens of dynamical systems and control theory. Specifically, we use contraction analysis to show that generalization and dynamical *robustness* are intimately related through the notion of algorithmic stability.

In particular, we prove that Riemannian contraction in a supervised learning setting implies generalization. We show that if a learning algorithm is contracting in some Riemannian metric with rate $\lambda > 0$, it is uniformly algorithmically stable with rate $\mathcal{O}(1/\lambda n)$, where n is the number of examples in the training set. The results hold for stochastic and deterministic optimization, in both continuous and discrete-time, for convex and non-convex loss surfaces. The associated generalization bounds reduce to well-known results in the particular case of gradient descent over convex or strongly convex loss surfaces. They can be shown to be optimal in certain linear settings, such as kernel ridge regression under gradient flow.

1 Introduction

Since the seminal work of Bousquet & Elisseeff (2002), the concept of *algorithmic stability* has been used to analyze the generalization properties of learning algorithms (Mukherjee et al., 2006; Shalev-Shwartz et al., 2009; 2010; Hardt et al., 2016). Roughly speaking, algorithmic stability refers to the notion that small changes to the training set will led to small changes in the output of the learning process.

In this work we focus on iterative optimizers within a supervised learning setting, where we are given access to a number of labelled training points drawn from some underlying common distribution, as well as a loss function which quantifies performance. Within this setting, we show that algorithmic stability is intimately related to notions of *robustness* from the dynamical systems and control literature. In particular, we make a connection between algorithmic stability and contractive stability (Lohmiller & Slotine, 1998). Loosely, a dynamical system is contracting if it forgets its initial conditions exponentially quickly.

Contraction analysis has found wide application in nonlinear control theory (Manchester & Slotine, 2017), robotics (Chung & Slotine, 2009), and synchronization (Pham & Slotine, 2007). However, it has only recently been applied to machine learning and neuroscience (Boffi et al., 2020; Kozachkov et al., 2020; Revay & Manchester, 2020; Jafarpour et al., 2021; Kozachkov et al., 2022; Centorrino et al., 2022; Burghi et al., 2022). We show that if an optimizer is *contracting* (Lohmiller & Slotine, 1998) in some Riemannian metric (in a precise sense defined below) then it is algorithmically stable. Due to the generality of contraction analysis and the flexibility afforded us by the choice of metric, our theory applies to wide variety of common optimizers–for example gradient flows and stochastic minibatch gradient descent–operating over both convex and non-convex loss surfaces (see Figures 1,2,and 3).

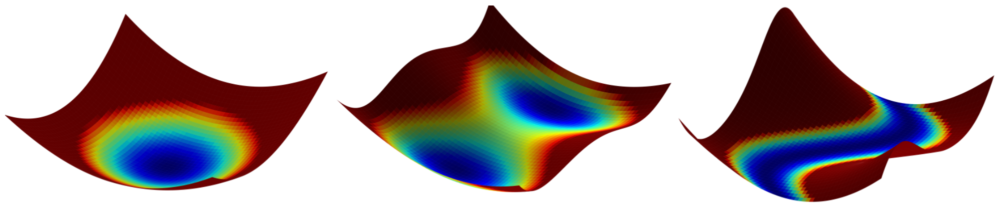

Figure 1: Example loss surfaces for which our results apply. Left panel: strongly convex and convex loss surfaces (sections 3.1 and 4.2.1). Middle panel: isolated local minima surrounded by basins of contraction (see Theorem 6). Right panel: valley of path-connected global minima (see section 4.2).

1.1 Related Work

A key early result in analyzing generalization in iterative optimization came from (Hardt et al., 2016), which established algorithmic stability for stochastic gradient methods. Later (Mou et al., 2018) proved similar results for stochastic gradient Langevin dynamics. Shortly thereafter (Charles & Papailiopoulos, 2018) showed that for loss functions satisfying certain geometrical constraints (e.g., the Polyak-Łojasiewicz inequality (Polyak, 1963)), any optimizer that converges to a global minimum is also algorithmically stable. Since then several follow-up works have analyzed the algorithmic stability of accelerated gradient methods, and the tradeoffs between optimization accuracy and algorithmic stability (Chen et al., 2018; Ho et al., 2020; Attia & Koren, 2021). The present work is similar in spirit to Charles & Papailiopoulos (2018), in the sense that we use an assumed stability property (in our case, contraction of optimizer trajectories) to derive generalization bounds for a wide class of optimizers. The following section introduces our supervised learning setting, which is the same setting as in Hardt et al. (2016), and provides necessary background on algorithmic stability.

1.2 Algorithmic Stability Background

We consider a generic supervised learning setting where we have access to n labelled examples, assumed to be drawn i.i.d from an unknown distribution $\mathcal{D}$ (Vapnik, 1999). We collect these examples into a training set $S = (z_1, \ldots, z_n)$. The *population risk* with respect to a loss function ℓ is defined as:

$$R[\boldsymbol{\theta}] = \mathbb{E}_{z\sim D}\ \ell(\boldsymbol{\theta}, z)$$

where $\boldsymbol{\theta} \in \mathbb{R}^m$ describes a model. We assume that we do not know the population risk, so we use the *empirical risk* as a proxy:

$$R_S[\boldsymbol{\theta}] = \frac{1}{n}\sum_{i=1}^{n}\ \ell(\boldsymbol{\theta}, z_i)$$

The difference between the population and empirical risk is denoted as the *generalization error* of model $\boldsymbol{\theta}$:

$$\Delta^{gen}(\boldsymbol{\theta}) \equiv R[\boldsymbol{\theta}] - R_S[\boldsymbol{\theta}]$$

We now define the stability of an algorithm, and relate it to this generalization error. Consider an algorithm $\mathcal{A}$ that takes in S and outputs a model (e.g., a parameter vector $\boldsymbol{\theta}$).

Definition 1.1 (Uniform Algorithmic Stability)**.** An algorithm $\mathcal{A}$ is ϵ-uniformly stable if for all data sets S, S' such that S and S' differ in at most one example, we have

$$\sup_z \mathbb{E}_{\mathcal{A}}[\ell(\mathcal{A}(S), z) - \ell(\mathcal{A}(S'), z)] \leq \epsilon \tag{1}$$

where the expectation is taken over the randomness of $\mathcal{A}$, if there is any. A fascinating result in learning theory states that uniform stability leads to generalization in expectation (Bousquet & Elisseeff, 2002; Shalev-Shwartz et al., 2010; Hardt et al., 2016). In particular we use Theorem 2.2 of Hardt et al. (2016).

Theorem 1. *Let $\mathcal{A}$ be ϵ-uniform stable and let $\mathbb{E}_{S,\mathcal{A}}$ denote an expectation taken over the samples S and the randomness of $\mathcal{A}$. Then, $|\mathbb{E}_{S,\mathcal{A}}\Big[\Delta^{gen}(\mathcal{A}(S))\Big]| \leq \epsilon$.*

If the output of $\mathcal{A}$ is some parameter vector $\boldsymbol{\theta}$ and we assume that our loss function is L-Lipshitz for every example z_i with respect to some norm $||\cdot||$, then the difference between two trajectories of an optimizer trained on set S and S' can be used to bound the generalization error, because:

$$\mathbb{E}_{\mathcal{A}}[|\ell(\boldsymbol{\theta}_S, z) - \ell(\boldsymbol{\theta}_{S'}, z)|] \leq L\,\mathbb{E}_{\mathcal{A}}||\boldsymbol{\theta}_S - \boldsymbol{\theta}_{S'}|| \tag{2}$$

Rather than only considering the Euclidean distance $||\boldsymbol{\theta}_S - \boldsymbol{\theta}_{S'}||$, in this paper we consider the *geodesic distance* $d_{\mathcal{M}}(\boldsymbol{\theta}_S, \boldsymbol{\theta}_{S'})$ computed on a Riemannian manifold $\mathcal{M} = (\mathbb{R}^m, \mathbf{M})$ (Figure 2). Here $\mathbf{M}(\boldsymbol{\theta}, t) \in \mathbb{R}^{m\times m}$ is the positive definite metric associated to $\mathcal{M}$. There are many optimization settings for which the geodesic distance between two points –as opposed to the Euclidean norm–is the more natural distance measure to consider (Amari, 1998; Wensing & Slotine, 2020). The main takeaway of this paper is that: *Riemannian contraction implies generalization in supervised learning.* The details about this generalization (e.g., its dependence on the number of samples n and the training time T) depend on the dynamical equations of the optimizer, as well as the geometry of the loss landscape, as we will see. We now provide background on nonlinear contraction analysis before stating our results.

1.3 Nonlinear Contraction Theory Background

Consider a state vector $\mathbf{x} \in \mathbb{R}^m$, evolving according to the continuous-time dynamics:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \tag{3}$$

where it is assumed that all quantities are real and smooth, so any required derivative or partial derivative exists and is continuous. Then we have the following definition:

Definition 1.2 (Contracting Dynamical System)**.** Denote the Jacobian of equation 3 by $\mathbf{J} \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}, t)$. If there exists a symmetric positive-definite metric $\mathbf{M}(\mathbf{x}, t) : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^{m\times m}$ and a scalar $\lambda > 0$ such that the following *differential Lyapunov equation* is uniformly satisfied in space and time:

$$\dot{\mathbf{M}} + \mathbf{MJ} + \mathbf{J}^T\mathbf{M} \leq -2\lambda\mathbf{M} \tag{4}$$

then the geodesic distance defined with respect to $\mathbf{M}$ between any two trajectories of equation 3 converges to zero exponentially, with rate λ, and equation 3 is said to be *contracting.* Discrete-time contraction can be defined similarly (Lohmiller & Slotine, 1998).

1.3.1 Robustness of Contracting Systems

Contracting systems are robust to disturbances, in the following sense. Assume that equation 3 is contracting in metric $\mathbf{M} = \mathbf{T}(\mathbf{x}, t)^T\mathbf{T}(\mathbf{x}, t)$ with rate λ. Now consider the same dynamics as equation 3, perturbed with some disturbance:

$$\dot{\mathbf{x}}_p = \mathbf{f}(\mathbf{x}_p, t) + \mathbf{d}(\mathbf{x}_p, t) \tag{5}$$

The geodesic distance $d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}_p)$ satisfies the *differential inequality*:

$$\frac{\mathrm{d}}{\mathrm{d}t} d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}_p) + \lambda d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}_p) \leq ||\mathbf{T}(\mathbf{x}, t)\mathbf{d}(\mathbf{x}_p, t)|| \tag{6}$$

Assuming there exists a finite constant D such that $||\mathbf{d}(\mathbf{x}_p, t)|| \leq D$ uniformly, equation 6 implies:

$$R(t) \leq \chi R(0)e^{-\lambda t} + \frac{D\chi}{\lambda} \tag{7}$$

where $R(t) \equiv ||\mathbf{x}(t) - \mathbf{x}_p(t)||$ and χ denotes an upper-bound on the condition number of $\mathbf{T}$. Likewise for the discrete-time dynamics contracting in some metric with rate $0 < \mu < 1$:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, t)$$

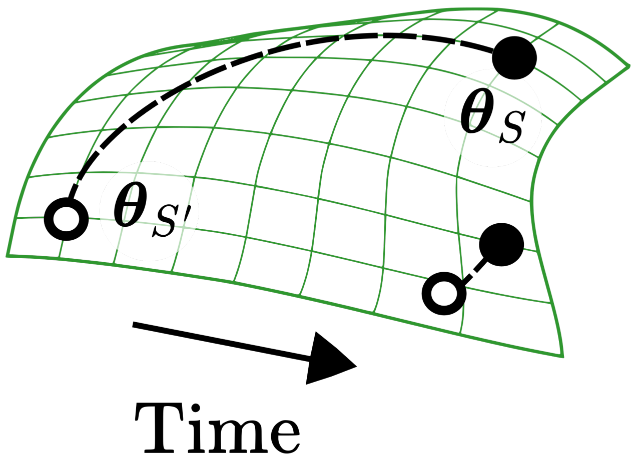

Figure 2: The geodesic distance between optimizer trajectories $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_{S'}$. If the optimizer is contracting with rate λ, this distance, denoted by the dashed line in the figure and $d_{\mathcal{M}}(\boldsymbol{\theta}_S, \boldsymbol{\theta}_{S'})$ in the text, shrinks until the two trajectories are within a ball of radius $\mathcal{O}(\frac{1}{n\lambda})$.

the analogous result is:

$$R(t) \leq \chi R(0)\mu^t + \frac{D\chi}{(1-\mu)} \tag{8}$$

For proofs of these statements we refer the reader to Lohmiller & Slotine (1998) (section 3.7, vii) as well as Del Vecchio & Slotine (2012) and Zhang et al. (2021) Proposition 1 in the appendix.

If we interpret equation 3 as an algorithm, then the only source of indeterminacy in this algorithm is the initial condition $\mathbf{x}(0)$. Therefore if equation 3 is always initialized within a ball of radius $C/2$ of some reference point, then equation 7 may be stated in expectation:

$$\mathbb{E}_{\mathcal{A}}[R(t)] \leq \mathbb{E}_{\mathcal{A}}[\chi R(0)e^{-\lambda t} + \frac{D\chi}{\lambda}] \leq \chi C e^{-\lambda t} + \frac{D\chi}{\lambda} \tag{9}$$

where we have used the linearity of the expectation value operator, as well as the assumption $\mathbb{E}_{\mathcal{A}}[R(0)] \leq C$.

1.3.2 Geodesics and Bounded Distortions

To ensure that our results are coordinate-free, we show that the 'distortion factor' between the geodesic distances computed along two different manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$ is uniformly bounded. The practical implication is that geodesic distances as measured in two different metrics can differ by no more than a constant factor, which precludes any situation where a system is stable in one metric (geodesic distances between trajectories shrink to zero) and not stable in another metric (geodesic distances do not shrink to zero).

Theorem 2. *Consider two Riemannian metrics $\mathbf{M}_1(\mathbf{x}, t)$ and $\mathbf{M}_2(\mathbf{x}, t)$ satisfying:*

$$\alpha_1 \mathbf{I} \preceq \mathbf{M}_1(\mathbf{x}, t) \preceq \beta_1 \mathbf{I} \tag{10}$$
$$\alpha_2 \mathbf{I} \preceq \mathbf{M}_2(\mathbf{x}, t) \preceq \beta_2 \mathbf{I} \tag{11}$$

with $\alpha_i, \beta_i > 0$. Then the corresponding geodesic distances evaluated between two points, $\mathbf{x}$ and $\mathbf{y}$, satisfy the bound:

$$\sqrt{\frac{\alpha_1}{\beta_2}} \leq \frac{d_{\mathcal{M}_1}(\mathbf{x}, \mathbf{y})}{d_{\mathcal{M}_2}(\mathbf{x}, \mathbf{y})} \leq \sqrt{\frac{\beta_1}{\alpha_2}}$$

2 Main Results

2.1 Contracting Optimizers are Algorithmically Stable

In this section we prove our main result for continuous-time optimizers using the entire training batch. We start with this case because it is the simplest. Later on, we provide the same result for stochastic, discrete-time optimizers such as mini-batch stochastic gradient descent. We assume that our parameter update is *sum-separable* with respect to training set S:

$$\dot{\boldsymbol{\theta}}_S = \mathbf{G}(\boldsymbol{\theta}_S, S) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{g}(\boldsymbol{\theta}_S, z_i) \tag{12}$$

In this case the output of algorithm $\mathcal{A}(S)$ is the vector $\boldsymbol{\theta}_S$ obtained by simulating equation 12 for time t. We also assume that $||\mathbf{g}|| \leq \xi$, for some constant ξ. If we interpret $\mathbf{g}$ as the gradient of some loss ℓ, then this corresponds to assuming that ℓ is ξ-Lipschitz. Finally, we assume that the optimizer is always initialized–perhaps randomly– within a ball of radius $C/2$ around some reference point. Now consider the same parameter update with respect to training set S', which differs from S in one example:

$$\dot{\boldsymbol{\theta}}_{S'} = \mathbf{G}(\boldsymbol{\theta}_{S'}, S') \tag{13}$$

We can now state our first main result:

Theorem 3. *[Contraction Implies Algorithmic Stability] If the dynamics equation 12 are contracting in metric $\mathbf{M} = \mathbf{T}(\boldsymbol{\theta}, t)^T\mathbf{T}(\boldsymbol{\theta}, t)$ with rate λ, then $\mathcal{A}$ is uniformly ϵ-stable, with:*

$$\epsilon \leq \chi L e^{-\lambda t} C + \frac{2\chi L\xi}{\lambda n} \tag{14}$$

where χ denotes a uniform upper-bound on the condition number of $\mathbf{T}(\boldsymbol{\theta}, t)$. Going forward we refer to $\epsilon_{stab} \equiv \frac{2\chi L\xi}{\lambda n}$.

Proof. The goal is to write equation 13 as a perturbed version of equation 12 and then apply the robustness property of contracting systems to yield the result. Letting k denote the index of the replaced element in $\mathcal{S}'$, observe that $\dot{\boldsymbol{\theta}}_{S'}$ may be written as follows:

$$\dot{\boldsymbol{\theta}}_{S'} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{g}(\boldsymbol{\theta}_{S'}, z_i) - \frac{1}{n}\big[\mathbf{g}(\boldsymbol{\theta}_{S'}, z_k) - \mathbf{g}(\boldsymbol{\theta}_{S'}, z'_k)\big]$$

where we have just subtracted out the term involving z_k from the sum, and added in the replacement term z'_k. This may be viewed as a perturbed version of equation 12, with disturbance:

$$||\mathbf{d}(\boldsymbol{\theta}_{S'}, z_k, z'_k, n)|| = ||\frac{1}{n}(\mathbf{g}(\boldsymbol{\theta}_{S'}, z_k) - \mathbf{g}(\boldsymbol{\theta}_{S'}, z'_k))|| \leq \frac{2\xi}{n} = D$$

Plugging D into equation 9, multiplying through by L because of equation 2, and taking the expectation $\mathbb{E}_{\mathcal{A}}$ to produce $R(0)$ yields the result. □

Remark 1 (Leave-One-Out Stability). *As pointed out in (Bousquet et al., 2020), for interpolation algorithms (such as, e.g., the highly overparameterized searches common in deep learning) it is more meaningful to analyze* leave-one-out *stability, rather than replace-one stability as we just did. In this case the same dynamical robustness argument applies immediately, so that D and therefore ϵ_{stab} are just reduced by a factor of two.*

Remark 2 (Generalization with High Probability). *A well-known limitation of using algorithmic stability to derive generalization bounds is that the bounds only hold in expectation. However, one can use Chebyshev's inequality to derive generalization bounds that hold with high probability (Bousquet & Elisseeff, 2002; Elisseeff et al., 2005; Feldman & Vondrak, 2019; Bousquet et al., 2020). It is well known that these bounds are tight in the case when algorithmic stability scales with $1/n$, see e.g., Theorem 12 and Remark 13 in (Bousquet & Elisseeff, 2002). Theorem 3 shows that (after exponentially decaying transients) deterministic contracting optimizers generalize with rate $1/n$. In Section 2.2, Theorem 4 will show that this $1/n$ scaling also holds for the stochastic optimization case.*

Remark 3 (Scaling Dynamics Does not Change Generalization Rate)**.** *Note that if we 'speed up' the dynamics equation 12 by some factor $\kappa > 0$:*

$$\mathbf{G}(\boldsymbol{\theta}_S, S) \rightarrow \kappa \mathbf{G}(\boldsymbol{\theta}_S, S)$$

one might intuitively expect the contraction rate to be scaled by κ as well ($\lambda \rightarrow \kappa\lambda$), which would allow an arbitrary increase of the rate of generalization in equation 14 by simply increasing κ. Note however that this is prevented by the presence of ξ in equation 14, which is also scaled by κ. The κ terms in the numerator and denominator therefore cancel out, leaving ϵ_{stab} unchanged. The exponentially decaying term in equation 14, however, decays with new rate $\kappa\lambda$.

Remark 4 (Lipschitz Assumption)**.** *As pointed out in Hardt et al. (2016), there are cases where L as defined in equation 2 may not exist. For example, strongly convex functions have unbounded gradients on $\mathbb{R}^m$. In this case we will overload the symbol L to be:*

$$L = \sup_{\boldsymbol{\theta} \in \Omega} \sup_{z} ||\nabla \ell(\boldsymbol{\theta}, z)||_2$$

where Ω denotes a compact set where the iterates of the optimizer are known to remain when initialized in a given compact region. For contracting optimizers and β-smooth loss functions ($\nabla^2 \ell \preceq \beta \mathbf{I}$), L is always finite. In particular, if we have some compact set of initial conditions and our optimizer is contracting (in a uniformly positive-definite metric), then the trajectories of the optimizer from any of those initial conditions will remain bounded. Indeed, any one trajectory will converge to a fixed point, and all the others must remain in a tube around its iterates (Lohmiller & Slotine, 1998). With this construction, we have a direct bound on the diameter of the set that the iterates of the optimizer must remain within, which we denote diameter(Ω). In this case we have $L \leq \beta$diameter(Ω).

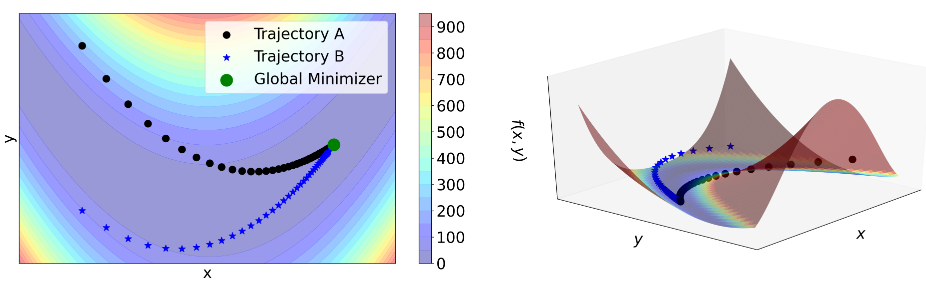

Figure 3: *Left subplot)* Two trajectories of a contracting optimizer, seeded from two different initial conditions, evolving over a non-convex loss surface (Rosenbrock function, $f(x, y) = 100(x^2 - y)^2 + (x - 1)^2$). Both exponentially converge to the global minimizer of the function. Trajectories superimposed over contour plot of the loss surface. *Right subplot)* A different view of the same optimization process, more clearly displaying the non-convexity of the loss surface.

2.2 Stochastic, Contracting Optimizers are Algorithmically Stable

In this section we show that a variant of Theorem 3 holds for stochastic, discrete-time optimizers (for example mini-batch stochastic gradient descent). Consider the iterative optimizer:

$$\boldsymbol{\theta}_{t+1}^S = \frac{1}{b} \sum_{i=1}^{b} \mathbf{g}(\boldsymbol{\theta}_t^S, z_i) \tag{15}$$

where $1 \leq b \leq n$ is the size of the mini-batch and z_i are samples drawn randomly from set S. As before we assume that $\mathbf{g}$ is smooth and bounded as $||\mathbf{g}|| \leq \xi$. Since equation 15 defines a discrete-time, random dynamical system (Tabareau & Slotine, 2013) we have to define what we mean by 'contraction'. In particular we will rely on an assumption of 'contraction in expectation', by which we mean the following. Consider two instantiations of the same discrete-time, random dynamical system:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, t, \Gamma)$$
$$\mathbf{y}_{t+1} = \mathbf{f}(\mathbf{y}_t, t, \Gamma)$$

with potentially different $\mathbf{x}_0$ and $\mathbf{y}_0$, and where Γ denotes a *particular realization* of a stochastic process which is the same for both $\mathbf{x}$ and $\mathbf{y}$. In our case, this stochasticity stems from the random sampling of training set datapoints to form a mini-batch. We will say that this system is *contracting in expectation* if for a sequence of metrics $\mathbf{M}_0, \ldots, \mathbf{M}_t$ we have:

$$\mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_{t+1}}(\mathbf{f}(\mathbf{x}_t, t, \Gamma), \mathbf{f}(\mathbf{y}_t, t, \Gamma))] \leq \mu \mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_t}(\mathbf{x}_t, \mathbf{y}_t)]$$

where $0 < \mu < 1$ and each metric is bounded $M_{min}\mathbf{I} \preceq \mathbf{M}_i \preceq M_{max}\mathbf{I}$. We can now state the following theorem:

Theorem 4. *[Contraction Implies Algorithmic Stability (Stochastic, Discrete)] Assume equation 15 is contracting in expectation, as defined above. In this case $\mathcal{A}$ is uniformly ϵ-stable with bound:*

$$\epsilon \leq L\chi C\mu^t + \frac{2\chi L\xi}{(1-\mu)n} \tag{16}$$

Proof. See appendix section A.1. □

3 Examples

3.1 Preconditioned Gradient Descent On Strongly Convex Loss Functions

In this example we show that our theory reproduces known stability bounds for gradient descent on strongly convex losses. To illustrate the role of the contraction metric, we consider *preconditioned* gradient descent. Consider this descent over an empirical loss function which is γ-strongly convex with respect to a parameter vector $\boldsymbol{\theta}$:

$$\dot{\boldsymbol{\theta}} = -\mathbf{P}^{-1}\nabla\mathcal{L}$$

where $\mathbf{P}$ is a positive-definite and symmetric matrix. Denote the largest and smallest eigenvalues of $\mathbf{P}$ as p_{max} and p_{min}, respectively. The Jacobian of this system is:

$$\mathbf{J} = \frac{\partial\dot{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}} = -\mathbf{P}^{-1}\nabla^2\mathcal{L}$$

Picking the metric $\mathbf{M} = \mathbf{P}$, we see that:

$$\mathbf{PJ} + \mathbf{J}^T\mathbf{P} = -2\nabla^2\mathcal{L} \leq -2\gamma\mathbf{I} \leq -\frac{2\gamma}{p_{max}}\mathbf{P}$$

and thus the system is contracting in metric $\mathbf{P}$ with rate $\lambda = \gamma/p_{max}$. Our algorithmic stability bound is therefore:

$$\epsilon_{stab} = \sqrt{\frac{p_{max}^3}{p_{min}}}\frac{2L^2}{\gamma n}$$

Where L is given by equation 4. Note that in the case of regular gradient descent, without preconditioning (i.e., $\mathbf{P} = \mathbf{I}$) the above analysis shows that $\lambda = \gamma$ and $\chi = 1$. Plugging these numbers into equation equation 14 yields the following:

$$\epsilon_{stab} = \frac{2L^2}{\gamma n}$$

which is precisely the result of Theorem 3.9 in (Hardt et al., 2016).

Remark 5 (Natural Gradient on Geodesically Strongly Convex Losses)**.** *Natural gradients are a popular way to incorporate geometric information about the loss surface into gradient-based optimization techniques (Amari, 1998; Zhang et al., 2019). An equivalence between g-Strong Convexity and global contraction of natural gradient flows was is given in (Theorem 1, (Wensing & Slotine, 2020)). That is, the optimizer dynamics:*

$$\dot{\boldsymbol{\theta}} = -\mathbf{M}(\boldsymbol{\theta})^{-1}\nabla\mathcal{L}(\boldsymbol{\theta})$$

are globally contracting if and only if $\mathcal{L}(\boldsymbol{\theta})$ *is geodesically strongly convex over* $\mathcal{M}$*. In this case Theorem 3 of the present work applies immediately, in precisely the same fashion as the preceding subsection.*

3.2 Picking the Best Metric for Kernel Regression

For constant metrics, our stability bound $\epsilon_{stab} \sim \frac{\chi}{\lambda}$ depends on the condition number of the contraction metric (specifically its square root) to the contraction rate *measured in that metric*. Different metrics yield different ϵ_{stab}, so it is natural to ask whether an 'optimal' metric $\mathbf{M}_{optimal}$ exists, such that:

$$\epsilon_{stab}(\mathbf{M}_{optimal}) \leq \epsilon_{stab}(\mathbf{M})$$

While finding such a metric is in general not easy to do, we show that it is possible in the case of gradient descent for kernel ridge regression (Shawe-Taylor et al., 2004). Kernel methods (which are inherently linear) can be used to derive insights into nonlinear systems such deep neural networks (Jacot et al., 2018; Lee et al., 2019; Fort et al., 2020; Canatar et al., 2021). Without loss of generality, we assume an element-wise feature map such that for a matrix $\mathbf{X} \in \mathbb{R}^{q \times z}$, the matrix $\phi(\mathbf{X}) \in \mathbb{R}^{q \times z}$ satisfies $\phi(\mathbf{X})_{ij} = \phi(\mathbf{X}_{ij})$. The squared-loss for kernel ridge-regression is:

$$\mathcal{L} = \frac{1}{2n}\sum_{i=1}^{n}(\phi(\mathbf{x}_i)\mathbf{w} - y_i)^2 + \frac{\alpha}{2}||\mathbf{w}||^2$$

where the $\phi(\mathbf{x}_i)$ are feature row vectors, $\mathbf{w} \in \mathbb{R}^m$ is the linear model to be learned, and the $y_i \in \mathbb{R}$ are target labels. The parameter $\alpha > 0$ is the regularization parameter. Under gradient descent $\dot{\mathbf{w}} = -\nabla\mathcal{L}$ the Jacobian of the optimizer dynamics is:

$$\frac{\partial\dot{\mathbf{w}}}{\partial\mathbf{w}} = \mathbf{J} = -(\mathbf{G} + \alpha\mathbf{I})$$

where $\mathbf{G} \equiv \frac{1}{n}\phi(\mathbf{X})^T\phi(\mathbf{X})$ and $\mathbf{X} \in \mathbb{R}^{n \times d}$ is a constant matrix with $\mathbf{x}_i$ as the i^{th} row. The Jacobian $\mathbf{J}$ is symmetric, constant and negative-definite. Thus, the optimizer is contracting in the identity metric with rate $\lambda_I = \lambda_{min}(\mathbf{G}) + \alpha$, where $\lambda_{min}(\cdot)$ denotes the smallest eigenvalue. We now prove the following:

Theorem 5. *For kernel ridge regression, the algorithmic stability bound* ϵ_{stab} *is minimized for* $\mathbf{M} = \mathbf{I}$.

Proof. Recall that for an arbitrary, constant metric we are looking for a positive-definite symmetric $\mathbf{Q}$ such that:

$$\mathbf{MJ} + \mathbf{JM} = -\mathbf{Q} \leq -2\lambda\mathbf{M}$$

Ignoring χ for a moment, we can ask: out of the set of all possible metrics, is there a metric that yields the *largest* contraction rate λ? An interesting result from linear dynamical systems theory is that the answer is in fact yes. While there can be many metrics for linear systems that give the largest possible λ, one can always be found from setting $\mathbf{Q} = \mathbf{I}$ and solving for $\mathbf{M}$ (see, e.g., section 3.5.5 in (Slotine et al., 1991)). Since $\mathbf{J}$ is symmetric, in our case this metric corresponds to the diagonalizing metric:

$$\mathbf{M}_{largest} = \frac{1}{2}\mathbf{J}^{-1} = \frac{1}{2}(\mathbf{G} + \alpha\mathbf{I})^{-1}$$

The contraction rate $\lambda_{largest}$ corresponding to this metric is:

$$\lambda_{largest} = \frac{1}{2}\frac{1}{\lambda_{max}(\mathbf{M}_{largest})} = \lambda_{min}(\mathbf{G}) + \alpha$$

which is precisely the same contraction rate as measured in the identity metric. Thus $\lambda_I = \lambda_{largest}$. Now we simply use the fact that $\chi_I = 1 \leq \chi_M$ for any metric. Since $\mathbf{M} = \mathbf{I}$ corresponds to the largest possible λ and the smallest possible $\chi = 1$, the ratio of χ to λ is minimal over all possible $\mathbf{M}$ when $\mathbf{M} = \mathbf{I}$. Thus:

$$\epsilon_{stab}(\mathbf{I}) \leq \epsilon_{stab}(\mathbf{M})$$

□

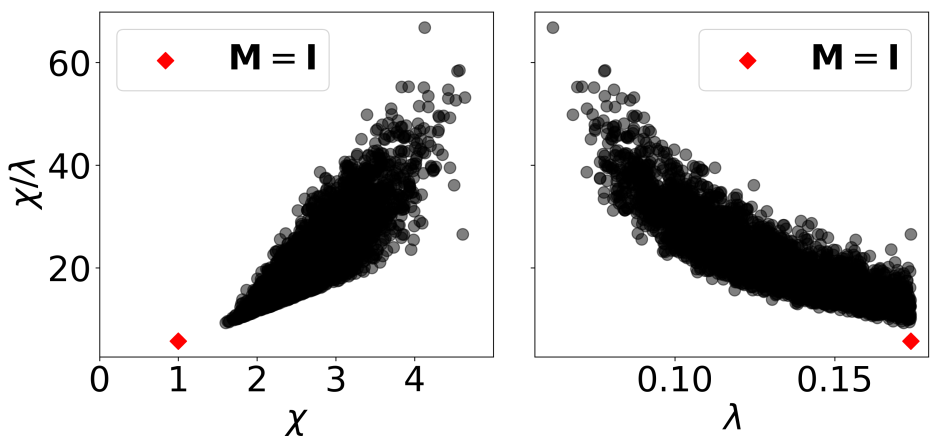

Figure 4: For a fixed $\mathbf{G} \in \mathbb{R}^{3\times3}$, we randomly generate $\mathbf{Q}$ and solve for $\mathbf{M}$. We then calculate the ratio χ/λ_M (the only metric-dependent terms in our algorithmic stability bound). We repeat this procedure 4000 times. *Left subplot)* The ratio χ/λ_M plotted against χ. *Right subplot)* The ratio χ/λ_M bound plotted against λ_M. Details are in main text. These plots illustrate our theoretical result: that the identity metric gives the optimal (i.e smallest) algorithmic stability bound. They also illustrates the reason: the identity metric simultaneously obtains the smallest condition number and the largest contraction rate, thus minimizing the ratio of the former to the latter.

This result is illustrated in Figure 4. To create this plot we generated a random $\mathbf{G} \in \mathbb{R}^{3\times3}$. Then we generated random $\mathbf{Q}$ and solved the Lyapunov equation for $\mathbf{M}$ using an implementation of the Bartels-Stewart algorithm in SciPy (Bartels & Stewart, 1972; Virtanen et al., 2020). In addition to these random $\mathbf{Q}$, we also set $\mathbf{Q} = \mathbf{I}$ to obtain the $\mathbf{M}$ corresponding to the largest λ. For each of these $\mathbf{Q}$ and $\mathbf{M}$ pairs, λ_M is given by $\lambda_M = \frac{1}{2}\frac{\lambda_{min}(\mathbf{Q})}{\lambda_{max}(\mathbf{M})}$ (Slotine et al., 1991). We computed $\mathbf{T}$ via a Cholesky decomposition (also using SciPy) and then performed a singular value decomposition to obtain χ. One interpretation of this result is: there is no 'better' coordinate system. That is, there is no coordinate transformation we could perform on the state vector $\mathbf{w}$ which would give us a tighter algorithmic stability bounds. This is because a constant metric $\mathbf{M} = \mathbf{T}^T\mathbf{T}$ corresponds to the coordinate change $\mathbf{w} \rightarrow \mathbf{T}\mathbf{w}$.

4 Weaker Notions of Stability

Contraction imposes a strong condition on optimizer trajectories: they must converge towards one another exponentially. Such convergence can be expected around isolated local or global minima, as discussed above. However in modern machine learning, one often observes optimizer trajectories which converge towards a common basin of low/zero loss, where minima may lie among a low-dimensional manifold (Garipov et al., 2018; Draxler et al., 2018; Fort et al., 2020; Liu et al., 2021). To accommodate these cases, we now discuss several weaker notions of contraction–specifically local contraction, semi-contraction and partial contraction–which also yield 'well-behaved' algorithmic stability bounds.

4.1 Loss Surfaces with Many Local Minima

A contraction region (i.e., a region of state space that satisfies Definition 1.2) for an autonomous system contains at most one equilibrium point (Lohmiller & Slotine, 1998). From this it follows that gradient descent over a loss surface with many local equilibria cannot be globally contracting. Fortunately, if Definition 1.2 holds within a subset of state-space, and additionally the system can be show to remain in that subset for all time (i.e., the subset is forward invariant), then that system is locally contracting. This motivates the following general result, as well as optimization-specific remark.

Theorem 6. *Consider the system equation 3 initialized inside an inner Euclidean ball of radius b, which is fully contained within an outer contraction region (which we also assume without loss of generality to be a Euclidean ball) of radius $B > b$. Assume that equation 3 stays within the inner ball for all time. Now consider the perturbed dynamics equation 5. If $B \geq b(\chi + 1) + \frac{\chi D}{\lambda}$ then equation 5 stays within the outer contraction region for all time, and the robustness result equation 7 holds.*

Proof. By equation 7, the perturbed trajectory will be at most distance $\chi b + \frac{\chi D}{\lambda}$ from the unperturbed trajectory. Since the unperturbed trajectory is always contained within a ball of radius b, this implies the perturbed trajectory is also contained in a ball of radius $b+\chi b+\frac{\chi D}{\lambda}$, assuming it stays in a contraction region. To ensure that it does in fact stay within a contraction region, we must have that $B \geq b(\chi + 1) + \frac{\chi D}{\lambda}$. □

Remark 6. *In the case of continuous-time optimizer equation 12, we have that $\boldsymbol{\theta}_S$ converges exponentially to the equilibrium point $\boldsymbol{\theta}_S^*$ enclosed by the contraction region:*

$$||\boldsymbol{\theta}^* - \boldsymbol{\theta}_S|| \leq \chi e^{-\lambda t} b$$

Since $\boldsymbol{\theta}_S^*$ is a particular trajectory of the optimizer dynamics, by robustness we also have:

$$||\boldsymbol{\theta}_S^* - \boldsymbol{\theta}_{S'}|| \leq \chi e^{-\lambda t} b + \frac{2\chi\xi}{\lambda n}$$

by the triangle inequality:

$$||\boldsymbol{\theta}_S - \boldsymbol{\theta}_{S'}|| \leq 2\chi e^{-\lambda t} b + \frac{2\chi\xi}{\lambda n}$$

this puts the following lower bound on the size of the contraction region B:

$$B > 2b\chi + \frac{2\chi\xi}{\lambda n}$$

4.2 Semi-Contracting Optimizers

If the optimizer is not strictly contracting ($\lambda > 0$), but instead is *semi-contracting* (i.e., $\lambda \geq 0$) then our algorithmic stability bound ϵ_{stab} is not independent of the training time. This is because the geodesic distance $d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}_p)$ between unperturbed and perturbed trajectories evolves according to:

$$\frac{\mathrm{d}}{\mathrm{d}t} d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}_p) + \lambda d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}_p) \leq ||\mathbf{T}(\mathbf{x}, t)\mathbf{d}(\mathbf{x}_p, t)||$$

If the only information we have about λ is that it is non-negative, then we can only bound the distance between trajectories as:

$$d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}_p) \leq \sup(||\mathbf{T}(\mathbf{x}, t)\mathbf{d}(\mathbf{x}_p, t)||)T + R(0)$$

Considering the disturbance bound $||\mathbf{d}(\mathbf{x}_p, t)|| \leq \frac{2\xi}{n}$ leads to the algorithmic stability bound:

$$\epsilon \leq L \left[\frac{2\chi\xi}{n} T + R(0) \right] \tag{17}$$

This bound holds generally for semi-contracting systems. However, without additional information about the optimizer dynamics, this bound gets worse as $T \to \infty$. If we know additional information about the dynamics–for example that they are modulated by a decaying learning rate–much tighter bounds can be obtained. We show this with the following example.

4.2.1 Example: Gradient Flows on Convex Losses

Here we show how the above analysis reproduces a well-known result from Hardt et al. (2016) regarding the algorithmic stability of SGD on convex (but not strongly convex) losses. We suppose that the Hessian of the loss function is positive semi-definite:

$$\nabla^2 \mathcal{L} \geq 0$$

Consider the gradient flow with learning rate scheduler (Goodfellow et al., 2016):

$$\dot{\boldsymbol{\theta}} = -\alpha(t)\nabla\mathcal{L}$$

where $\alpha(t) \geq 0$. This optimizer is semi-contracting in the identity metric, since:

$$\frac{\partial \dot{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} = -\alpha(t)\nabla^2\mathcal{L} \leq 0$$

In this case the disturbance term in Theorem 3 is the same as before, just with an addition $\alpha(t)$ factored in. To facilitate comparison with Hardt et al. (2016), we assume as they do that the optimizer is always initialized at the origin (i.e., $R(0) = 0$). The Euclidean distance between the optimizer trajectories on training sets S and S' evolves according to:

$$\dot{R} \leq \alpha(t)D$$

where $D = 2L/n$. Integrating this inequality and setting $R(0)$ yields the algorithmic stability bound:

$$\epsilon \leq \frac{2L^2}{n}\int_{t=0}^{T} \alpha(t)dt$$

which is the result of Hardt et al. (2016), Theorem 3.8. We remind the reader that the extra factor of L is picked up from equation 2. This result helps explain why a decaying learning rate is a useful strategy in deep learning–if the learning rate decays quickly enough (e.g., exponentially), the above integral converges, so that n and T do not compete with each other, as they do in equation 17.

Remark 7. *The equivalence between semi-contraction of natural gradient flows and geodesic convexity was recently proven in Wensing & Slotine (2020). Thus the above algorithmic stability bound extends immediately to this case.*

4.3 Partial Contraction

In many cases of interest, a 'pure' contraction analysis is hard or difficult to do. For example when an optimizer has an adaptive learning rate, this can significantly complicate the calculation of the Jacobian. To deal with these difficulties, we make use of a generalization of contraction introduced in Wang & Slotine (2005), known as *partial contraction*.

Definition 4.1 (Partially Contracting Dynamical System)**.** Consider the system equation 3 (not nessesarily contracting) and an auxiliary system of the form:

$$\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}, \mathbf{x}, t)$$

We assume that this auxiliary system for $\mathbf{y}$ is contracting in metric $\mathbf{M}$ with rate λ. We also assume that $\mathbf{g}(\mathbf{x}, \mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t)$. If a single, particular trajectory $\mathbf{y}(t)$ of the auxiliary system is known, then all trajectories of equation 3 converge exponentially towards $\mathbf{y}(t)$. In this case we say that equation 3 is partially contracting.

Partially contracting systems are also robust to disturbances (Del Vecchio & Slotine, 2012), a property we will make use of. In particular we have the following theorem:

Theorem 7. *[Robustness of Partially Contracting Systems] Assume that equation 3 is partially contracting, and now perturb it with some disturbance $||\mathbf{d}(\mathbf{x}, t)|| \leq D$:*

$$\dot{\mathbf{x}}_p = \mathbf{f}(\mathbf{x}_p, t) + \mathbf{d}(\mathbf{x}_p, t)$$

Then after exponential transients of rate λ, we have the following robustness result:

$$||\mathbf{x} - \mathbf{x}_p|| \leq \frac{D\chi}{\lambda}$$

4.3.1 Adaptive Learning Rates

The above results can be extended to include the presence of a state-dependent, time-varying, learning rate (Zeiler, 2012; Kingma & Ba, 2014; Goodfellow et al., 2016; Liu et al., 2019). In particular consider the learning dynamics with a learning rate scheduler $\rho(\boldsymbol{\theta}, t)$:

$$\dot{\boldsymbol{\theta}}_S = -\rho(\boldsymbol{\theta}_S, t)\mathbf{G}(\boldsymbol{\theta}_S, S)$$

where $\rho_{max} \geq \rho(\boldsymbol{\theta}, t) \geq \rho_{min} > 0$. Now consider the auxiliary *virtual system*:

$$\dot{\boldsymbol{\theta}}_y = -\rho(\boldsymbol{\theta}_S, t)\mathbf{G}(\boldsymbol{\theta}_y, S)$$

By Theorem 7, if the $\boldsymbol{\theta}_y$ system is contracting in $\mathbf{M}$ with rate $\rho_{min}\lambda$, then we have, by the same arguments in Theorem 3, that $\mathcal{A}(S)$ is asymptotically (after exponential transients of rate $\rho_{min}\lambda$) uniformly ϵ-stable with:

$$\epsilon_{stab} = \frac{2\chi L \rho_{max} \xi}{\rho_{min} \lambda n}$$

5 Concluding Remarks and Comments

5.1 Comparison to Related Work

Our results are similar in spirit to Charles & Papailiopoulos (2018), in the sense that we also use an optimizer's intrinsic dynamical stability to provide generalization error bounds. In certain cases–for example gradient flow on strongly convex losses–our results allow us to derive tighter bounds, because we do not assume the existence of a global minimizer $\boldsymbol{\theta}^*$ and go through the triangle inequality to bound the distance between $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_{S'}$.

5.2 Future Directions

Contracting systems are robust to noise (Pham & Slotine, 2013), and therefore it seems likely that the results presented here can straightfowardly be extended to stochastic gradient flows–along the lines of Mandt et al. (2015) or Boffi & Slotine (2020). Furthermore, we only examined generalization error here and did not analyze the bias-variance trade-off. Our work also suggests a potential connection to the double descent phenomenon (Nakkiran et al., 2021). In particular equation 14 implies that the generalization error can overshoot by a factor of $L\chi$, which gives room for the generalization to increase transiently from its initial value before it eventually decreases. This will be explored in future work.

We conclude with some speculations on how the above results relate to biology, specifically neuroscience. The role that non-Euclidean geometry plays in objective-based functions of the brain is an interesting and open question (Surace et al., 2020). Many local synaptic rules can be thought of as implementing optimization over a loss function. For example in certain settings Hebbian plasticity minimizes Principal Component Loss (Oja, 1992). It seems plausible that our results may be used to quantify the generalization behavior of such rules. We also did not explore the combination properties of contracting systems. Contracting systems can be combined in various forms of hierarchy and feedback in ways which automatically preserve contraction (Lohmiller & Slotine, 1998; Slotine & Lohmiller, 2001; Kozachkov et al., 2021). The results here suggest that combinations of contracting optimizers automatically generalize well–which is a property one could easily imagine evolution would like to preserve in a system like the brain.

References

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Amit Attia and Tomer Koren. Algorithmic instabilities of accelerated gradient descent. *Advances in Neural Information Processing Systems*, 34, 2021.

R. H. Bartels and G. W. Stewart. Solution of the matrix equation ax + xb = c [f4]. *Commun. ACM*, 15 (9):820–826, sep 1972. ISSN 0001-0782. doi: 10.1145/361573.361582. URL `https://doi.org/10.1145/361573.361582`.

Nicholas M Boffi and Jean-Jacques E Slotine. A continuous-time analysis of distributed stochastic gradient. *Neural computation*, 32(1):36–96, 2020.

Nicholas M Boffi, Stephen Tu, Nikolai Matni, Jean-Jacques E Slotine, and Vikas Sindhwani. Learning stability certificates from data. *CoRL*, 2020.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Olivier Bousquet, Yegor Klochkov, and Nikita Zhivotovskiy. Sharper bounds for uniformly stable algorithms. In *Conference on Learning Theory*, pp. 610–626. PMLR, 2020.

Thiago B Burghi, Timothy O'Leary, and Rodolphe Sepulchre. Distributed online estimation of biophysical neural networks. *arXiv preprint arXiv:2204.01472*, 2022.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications*, 12(1): 1–12, 2021.

Veronica Centorrino, Francesco Bullo, and Giovanni Russo. Contraction analysis of hopfield neural networks with hebbian learning. *arXiv preprint arXiv:2204.05382*, 2022.

Zachary Charles and Dimitris Papailiopoulos. Stability and generalization of learning algorithms that converge to global optima. In *International Conference on Machine Learning*, pp. 745–754. PMLR, 2018.

Yuansi Chen, Chi Jin, and Bin Yu. Stability and convergence trade-off of iterative optimization algorithms. *arXiv preprint arXiv:1804.01619*, 2018.

Soon-Jo Chung and Jean-Jacques E Slotine. Cooperative robot control and concurrent synchronization of lagrangian systems. *IEEE transactions on Robotics*, 25(3):686–700, 2009.

Domitilla Del Vecchio and Jean-Jacques E Slotine. A contraction theory approach to singularly perturbed systems. *IEEE Transactions on Automatic Control*, 58(3):752–757, 2012.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.

Andre Elisseeff, Theodoros Evgeniou, Massimiliano Pontil, and Leslie Pack Kaelbing. Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6(1), 2005.

Vitaly Feldman and Jan Vondrak. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conference on Learning Theory*, pp. 1270–1279. PMLR, 2019.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *arXiv preprint arXiv:2010.15110*, 2020.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8803–8812, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent, 2016.

Nhat Ho, Koulik Khamaru, Raaz Dwivedi, Martin J Wainwright, Michael I Jordan, and Bin Yu. Instability, computational efficiency and statistical accuracy. *arXiv preprint arXiv:2005.11411*, 2020.

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

Saber Jafarpour, Alexander Davydov, Anton Proskurnikov, and Francesco Bullo. Robust implicit networks via non-euclidean contractions. *Advances in Neural Information Processing Systems*, 34:9857–9868, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Leo Kozachkov, Mikael Lundqvist, Jean-Jacques Slotine, and Earl K Miller. Achieving stable dynamics in neural circuits. *PLoS computational biology*, 16(8):e1007659, 2020.

Leo Kozachkov, Michaela Ennis, and Jean-Jacques Slotine. Recursive construction of stable assemblies of recurrent neural networks. *arXiv preprint arXiv:2106.08928*, 2021.

Leo Kozachkov, John Tauber, Mikael Lundqvist, Scott L Brincat, Jean-Jacques Slotine, and Earl K Miller. Robust working memory through short-term synaptic plasticity. *bioRxiv*, 2022.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32:8572–8583, 2019.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks, 2021.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

Ian R Manchester and Jean-Jacques E Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Transactions on Automatic Control*, 62(6):3046–3053, 2017.

Stephan Mandt, Matthew D Hoffman, David M Blei, et al. Continuous-time limit of stochastic gradient descent revisited. In *OPT workshop, NIPS*, 2015.

Wenlong Mou, Liwei Wang, Xiyu Zhai, and Kai Zheng. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. In *Conference on Learning Theory*, pp. 605–638. PMLR, 2018.

Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25(1):161–193, 2006.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Erkki Oja. Principal components, minor components, and linear neural networks. *Neural networks*, 5(6): 927–935, 1992.

Quang-Cuong Pham and Jean-Jacques Slotine. Stable concurrent synchronization in dynamic system networks. *Neural networks*, 20(1):62–77, 2007.

Quang-Cuong Pham and Jean-Jacques Slotine. Stochastic contraction in riemannian metrics. *arXiv preprint arXiv:1304.0340*, 2013.

Boris Polyak. Gradient methods for the minimisation of functionals. *Ussr Computational Mathematics and Mathematical Physics*, 3:864–878, 1963.

Max Revay and Ian Manchester. Contracting implicit recurrent neural networks: Stable models with improved trainability. In *Learning for Dynamics and Control*, pp. 393–403. PMLR, 2020.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT*, volume 2, pp. 5, 2009.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670, 2010.

John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

J.-J.E. Slotine and W. Lohmiller. Modularity, evolution, and the binding problem: a view from stability theory. *Neural Networks*, 14(2):137–145, 2001. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(00)00089-7. URL `https://www.sciencedirect.com/science/article/pii/S0893608000000897`.

Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.

Simone Carlo Surace, Jean-Pascal Pfister, Wulfram Gerstner, and Johanni Brea. On the choice of metric in gradient-based theories of brain function. *PLoS computational biology*, 16(4):e1007640, 2020.

Nicolas Tabareau and Jean-Jacques Slotine. Contraction analysis of nonlinear random dynamical systems. *arXiv preprint arXiv:1309.5317*, 2013.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Wei Wang and Jean-Jacques E Slotine. On partial contraction analysis for coupled nonlinear oscillators. *Biological cybernetics*, 92(1):38–53, 2005.

Patrick M Wensing and Jean-Jacques Slotine. Beyond convexity—contraction and global convergence of gradient descent. *Plos one*, 15(8):e0236661, 2020.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Guodong Zhang, James Martens, and Roger Grosse. Fast convergence of natural gradient descent for overparameterized neural networks. *arXiv preprint arXiv:1905.10961*, 2019.

Thomas TCK Zhang, Stephen Tu, Nicholas M Boffi, Jean-Jacques E Slotine, and Nikolai Matni. Adversarially robust stability certificates can be sample-efficient. *arXiv preprint arXiv:2112.10690*, 2021.

A Appendix

A.1 Proof of equation 8

Proof. For every time t we randomly sample b indices $i_1, \cdots, i_b$. Using these indices we select datapoints $z_i, \cdots, z_{i_b}$ from S and S' to update $\boldsymbol{\theta}_t^S$ and $\boldsymbol{\theta}_t^{S'}$ respectively. At every time t there are two possibilities. Either we do not draw the replaced element z'_k or we do. Denote these events A and B, respectively (Figure 5). We have $P(A) = 1 - \frac{b}{n}$ and $P(B) = \frac{b}{n}$. If event A occurs, then by assumption we expect the geodesic distance to shrink:

$$\mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_{t+1}}(\boldsymbol{\theta}_{t+1}^S, \boldsymbol{\theta}_{t+1}^{S'})|A] \leq \mu \mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_t}(\boldsymbol{\theta}_t^S, \boldsymbol{\theta}_t^{S'})|A] \tag{18}$$

where $\mathbb{E}[\cdot|A]$ denotes the conditional expectation given event A. However, if the replaced element is drawn (i.e., event B occurs) then we have:

$$\boldsymbol{\theta}_{t+1}^S = \frac{1}{b}\sum_{i=1}^{b} \mathbf{g}(\boldsymbol{\theta}_t^S, z_i) \equiv \hat{\mathbf{G}}(\boldsymbol{\theta}_t^S)$$

$$\boldsymbol{\theta}_{t+1}^{S'} = \hat{\mathbf{G}}(\boldsymbol{\theta}_t^{S'}) + \mathbf{d}(\boldsymbol{\theta}_t^{S'})$$

where $\mathbf{d}(\boldsymbol{\theta}_t^{S'}) = \frac{1}{b}(\mathbf{g}(\boldsymbol{\theta}_t^{S'}, z'_k) - \mathbf{g}(\boldsymbol{\theta}_t^{S'}, z_k))$. As in Theorem equation 3, we have written the update for $\boldsymbol{\theta}^{S'}$ as a 'perturbed' version of the update for $\boldsymbol{\theta}^S$. We will now derive an analogous robustness result, and then use the linearity of expectation to bound the overall geodesic distance. Note that:

$$\begin{aligned}
d_{\mathcal{M}_{t+1}}(\boldsymbol{\theta}_{t+1}^S, \boldsymbol{\theta}_{t+1}^{S'}) &= d_{\mathcal{M}_{t+1}}(\hat{\mathbf{G}}(\boldsymbol{\theta}_t^S), \hat{\mathbf{G}}(\boldsymbol{\theta}_t^{S'}) + \mathbf{d}(\boldsymbol{\theta}_t^{S'})) \\
&\leq d_{\mathcal{M}_{t+1}}(\hat{\mathbf{G}}(\boldsymbol{\theta}_t^S), \hat{\mathbf{G}}(\boldsymbol{\theta}_t^{S'})) + d_{\mathcal{M}_{t+1}}(\hat{\mathbf{G}}(\boldsymbol{\theta}_t^{S'}), \hat{\mathbf{G}}(\boldsymbol{\theta}_t^{S'}) + \mathbf{d}(\boldsymbol{\theta}_t^{S'})) \\
&\leq d_{\mathcal{M}_{t+1}}(\hat{\mathbf{G}}(\boldsymbol{\theta}_t^S), \hat{\mathbf{G}}(\boldsymbol{\theta}_t^{S'})) + \sqrt{M_{max}}\frac{2\xi}{b}
\end{aligned}$$

where the first inequality comes from the triangle inequality and the second comes from the boundedness of $\mathbf{d}(\boldsymbol{\theta}_t^{S'})$ and the metric distortion bound in Theorem 2. Now applying the assumption of contraction in expectation we get:

$$\begin{aligned}
\mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_{t+1}}(\boldsymbol{\theta}_{t+1}^S, \boldsymbol{\theta}_{t+1}^{S'})|B] &\leq \mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_{t+1}}(\hat{\mathbf{G}}(\boldsymbol{\theta}_t^S), \hat{\mathbf{G}}(\boldsymbol{\theta}_t^{S'}))|B] + \sqrt{M_{max}}\frac{2\xi}{b} \\
&\leq \mu \mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_t}(\boldsymbol{\theta}_t^S, \boldsymbol{\theta}_t^{S'})|B] + \sqrt{M_{max}}\frac{2\xi}{b}
\end{aligned} \tag{19}$$

We can now use the linearity of the expectation operator to bound the geodesic distance, and then use the metric distortion result to bound the Euclidean distance:

$$\begin{aligned}
\mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_{t+1}}(\boldsymbol{\theta}_{t+1}^S, \boldsymbol{\theta}_{t+1}^{S'})] &= \mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_{t+1}}(\boldsymbol{\theta}_{t+1}^S, \boldsymbol{\theta}_{t+1}^{S'})|A]P(A) + \mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_{t+1}}(\boldsymbol{\theta}_{t+1}^S, \boldsymbol{\theta}_{t+1}^{S'})|B]P(B) \\
&\leq \mu \mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_t}(\boldsymbol{\theta}_t^S, \boldsymbol{\theta}_t^{S'})](1 - \frac{b}{n}) + (\mu \mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_t}(\boldsymbol{\theta}_t^S, \boldsymbol{\theta}_t^{S'})] + \sqrt{M_{max}}\frac{2\xi}{b})\frac{b}{n} \\
&= \mu \mathbb{E}_{\mathcal{A}}[d_{\mathcal{M}_t}(\boldsymbol{\theta}_t^S, \boldsymbol{\theta}_t^{S'})] + \sqrt{M_{max}}\frac{2\xi}{n}
\end{aligned}$$

Using the metric distortion bounds and unravelling the recursion yields:

$$\mathbb{E}_{\mathcal{A}}[d(\boldsymbol{\theta}_t^S, \boldsymbol{\theta}_t^{S'})] \leq \chi \mu^t C + \frac{2\chi\xi}{(1-\mu)n}$$

Where $\chi = \sqrt{\frac{M_{max}}{M_{min}}}$ has again come from the metric distortion bound. Multiplying through by L, we have that:

$$\epsilon_{stab} = \frac{2L\chi\xi}{n(1-\mu)}$$

which is the same result as the continuous-time case, expect that $\lambda \to (1-\mu)$. $\square$

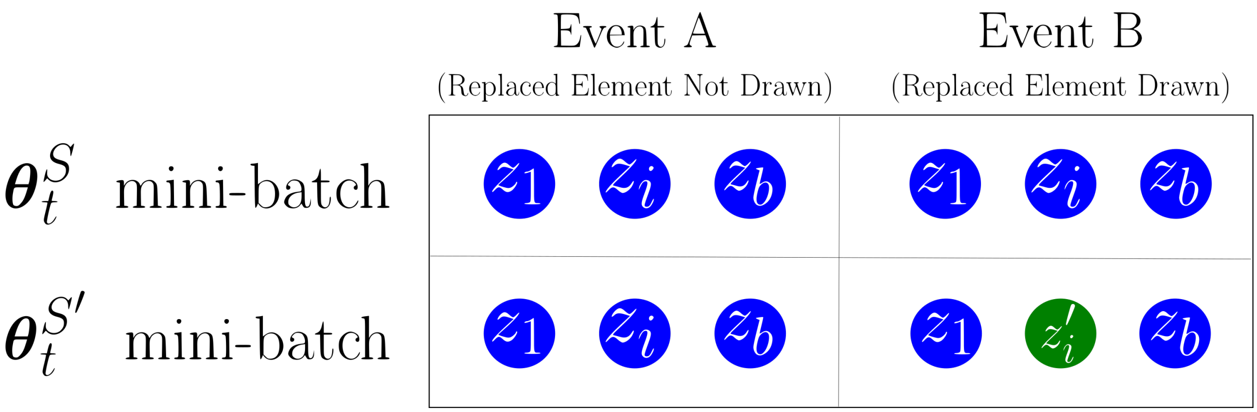

Figure 5: Illustration of the two cases for updating with mini-batches. At each time t, we randomly sample b indices between 1 and n. Then we draw the corresponding data-points from sets S and S' to form the mini-batches used to update $\boldsymbol{\theta}_t^S$ and $\boldsymbol{\theta}_t^{S'}$ respectively. In Event A (left column), the index of the replaced element is not selected, and therefore the datapoints used to update $\boldsymbol{\theta}_t^S$ and $\boldsymbol{\theta}_t^{S'}$ are the same. In Event B, the index of the replaced element is selected, and so the datapoints used to perform the update are different.