Curvature-corrected learning dynamics in deep neural networks

Dongsung Huh [1]

Abstract

Deep neural networks exhibit complex learning dynamics due to their non-convex loss landscapes. Second-order optimization methods facilitate learning dynamics by compensating for ill-conditioned curvature. In this work, we investigate how curvature correction modifies the learning dynamics in deep linear neural networks and provide analytical solutions. We derive a generalized conservation law that preserves the path of parameter dynamics from curvature correction, which shows that curvature correction only modifies the temporal profiles of dynamics along the path. We show that while curvature correction accelerates the convergence dynamics of the input-output map, it can also negatively affect the generalization performance. Our analysis also reveals an undesirable effect of curvature correction that compromises stability of parameters dynamics during learning, especially with block-diagonal approximation of natural gradient descent. We introduce fractional curvature correction that resolves this problem while retaining most of the acceleration benefits of full curvature correction.

1. Introduction

The non-convex loss landscapes of deep neural networks exhibit ill-conditioned curvature and saddle-points where gradient-based first-order optimization methods can perform poorly (Martens, 2010; Dauphin et al., 2014), which produces complex nonlinear learning dynamics (Saxe et al., 2013). Second order methods, such as natural gradient descent (NGD) (Amari, 1998), compensate for the effect of curvature by using the distance metric intrinsic to the space of input-output functions (Pascanu & Bengio, 2013; Martens, 2014; Botev et al., 2017; Bernacchia et al., 2018). While recent advances in approximate NGD methods (e.g. K-FAC) have dramatically improved the computational efficiency for practical scale applications (Ba et al., 2016; Grosse & Martens, 2016; Botev et al., 2017; Martens et al., 2018; Osawa et al., 2019), however, it remains largely unknown how curvature correction actually modifies the learning dynamics in deep networks. Do the curvature-corrected learning rule simply accelerate convergences towards the same minimum solutions as gradient descent, or do they impose bias toward qualitatively different solutions?

As a first step toward establishing theoretical understanding of these questions, we analyze the learning dynamics of deep linear networks under a spectrum of curvature-corrected update rules. Deep linear networks provide an excellent mathematical framework for developing theoretical insights on the complex inner workings of deep nonlinear networks (Goodfellow et al., 2016). Despite their simplicity, deep linear networks capture the essential nonlinear relationship between network's input-output maps and their parameters. Recently, many works have analyzed the learning trajectories of deep linear networks under gradient descent to compute the convergence rate under various initial conditions and architectures (Arora et al., 2018a;b; Bartlett et al., 2019; Du & Hu, 2019), revealed decoupled modes of convergence dynamics to explain the origin of multiple stage-like loss profiles (Saxe et al., 2013), and showed implicit biases for regularization (Du et al., 2018; Arora et al., 2019) and resistance to overfitting (Advani & Saxe, 2017; Lampinen & Ganguli, 2018; Poggio et al., 2018). Yet, it is unknown how these properties of convergence dynamics generalize beyond the first-order update rules.

Our contribution The main results are summarized as follows.

1. We show that the path of parameter dynamics is preserved under curvature correction by deriving a generalized conservation law that dictates the path shape. Consequently, curvature correction only affects the temporal profile of dynamics along the paths.

2. We show a trade-off between the accelerated dynamics of network's input-output map and the stability of parameter dynamics during learning: The process of full curvature correction, which completely removes the

[1]MIT-IBM Watson AI Lab, Cambridge, Massachusetts, USA. Correspondence to: Dongsung Huh <huh@ibm.com>.

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

non-linearity of map dynamics, produces exploding parameter dynamics at saddle points.

3. We introduce a fractional curvature-corrected update rule called $\sqrt{\text{NGD}}$, which resolves the vanishing/exploding speed problems of SGD/NGD. This makes the map dynamics moderately nonlinear, but no more so than that of one-hidden-layer networks under gradient descent.

4. The widely-used block-diagonal approximations of NGD breaches the aforementioned conservation law, and results in highly divergent parameter update dynamics. In contrast, block-diagonalization of $\sqrt{\text{NGD}}$ preserves the stability of parameter update dynamics, yielding efficient and stable learning algorithms.

5. NGD makes the learning dynamics prone to overfitting by simultaneously learning the signal and the noise dimensions of data. In contrast, $\sqrt{\text{NGD}}$ retains the gradient descent's resistance to overfitting by preferentially learning the signal dimensions first before learning the noise dimensions.

2. Setup and notations

Consider a linear neural network of depth d, whose parameters are the weight matrices $\boldsymbol{w} \equiv \{w_i\}_{i=1}^d$. The network's input-output map is given by the total weight $\bar{w} \equiv \prod_{i=1}^d w_i = w_d \cdots w_1$, such that $f_{\boldsymbol{w}}(x) = \bar{w}x = \hat{y}$, which learns the statistics of a dataset $D = \{x^\mu, y^\mu\}_{\mu=1}^P$ by minimizing the l_2 loss

$$L(\boldsymbol{w}) = \mathbb{E}_D\left[\frac{1}{2}\|\bar{w}x - y\|^2\right] = \text{Tr}\left[\frac{1}{2}\Delta\Sigma_x\Delta^\intercal\right] + const, \tag{1}$$

where $\mathbb{E}_D$ is the expectation over dataset D, $\Sigma_x \equiv \mathbb{E}_D[xx^\intercal]$ is the input correlations, $\bar{w}_* \equiv \mathbb{E}_D[yx^\intercal]\,\Sigma_x^{-1}$ is the desired map, and $\Delta \equiv \bar{w} - \bar{w}_*$ denotes the displacement between $\bar{w}$ and $\bar{w}_*$. For the ease of exposition, we consider pre-whitened input dataset such that $\Sigma_x = I$.

Gradient and Hessian$_+$ We use array representations and bold symbols to denote the derivatives of network parameters: For example, gradient descent is expressed as $\dot{\boldsymbol{w}} = -\eta\boldsymbol{g}$, where $\dot{\boldsymbol{w}} = \begin{bmatrix}\dot{w}_1 \\ \dot{w}_2\end{bmatrix}$ and $\boldsymbol{g} = \begin{bmatrix}w_2^\intercal\,\Delta \\ \Delta\,w_1^\intercal\end{bmatrix}$ represent the continuous-time weight update and the gradient of a depth $d = 2$ network.

In vectorized notations, gradient and Hessian of eq (1) can be expressed as $\boldsymbol{g} = \boldsymbol{J}\Delta$ and $\boldsymbol{H} = \boldsymbol{J}\boldsymbol{J}^\intercal + \boldsymbol{J}'\Delta$, where $\boldsymbol{J}$ is the Jacobian tensor, *i.e.* the derivative of the input-output map $J_i \equiv \partial\bar{w}/\partial w_i$, and $\boldsymbol{J}'$ is the second derivative.

Hessian operates on weight update to produce the gradient update (*i.e.* Hessian-vector product):

$$\boldsymbol{H}\dot{\boldsymbol{w}} = \dot{\boldsymbol{g}} = \begin{bmatrix}w_2^\intercal\dot{\Delta} + \dot{w}_2^\intercal\Delta \\ \dot{\Delta}w_1^\intercal + \Delta\dot{w}_1^\intercal\end{bmatrix}. \tag{2}$$

Most second order methods use positive semi-definite (PSD) approximations of Hessian (*e.g.* Fisher matrix (Amari, 1998; Heskes, 2000; Martens & Grosse, 2015), Generalized-Gauss-Newton matrix (Martens, 2014; Botev et al., 2017)) to guarantee convergence to local minima. This corresponds to discarding the second term of Hessian, *i.e.* $\boldsymbol{H}_+ = \boldsymbol{J}\boldsymbol{J}^\intercal$, whose operation on weight update is

$$\boldsymbol{H}_+\dot{\boldsymbol{w}} = \boldsymbol{J}\boldsymbol{J}^\intercal\dot{\boldsymbol{w}} = \begin{bmatrix}w_2^\intercal\dot{\Delta} \\ \dot{\Delta}\,w_1^\intercal\end{bmatrix}, \tag{3}$$

since $\boldsymbol{J}^\intercal\dot{\boldsymbol{w}} = \dot{\bar{w}} = \dot{\Delta}$. $\boldsymbol{H}_+$ is indeed PSD, since $\dot{\boldsymbol{w}}\cdot\boldsymbol{H}_+\dot{\boldsymbol{w}} = \text{Tr}[\dot{\Delta}\dot{\Delta}^\intercal] \geq 0$ (Dot-product: $\boldsymbol{a}\cdot\boldsymbol{b} \equiv \sum_{i=1}^d \text{Tr}[a_i b_i^\intercal]$).

Symmetries and Null-updates Deep linear networks exhibit inherent symmetries that the input-output map $\bar{w}$ is invariant under transformations that multiply an arbitrary square matrix m to one layer and its inverse to the next: $\begin{bmatrix}w_1 \\ w_2\end{bmatrix} \to \begin{bmatrix}mw_1 \\ w_2m^{-1}\end{bmatrix}$. Equivalently, $\dot{\boldsymbol{w}}_\varnothing \equiv \begin{bmatrix}mw_1 \\ -w_2m\end{bmatrix}$ are the continuous-time transformations that yield the invariance $\dot{\bar{w}} = w_2\dot{w}_{\varnothing 1} + \dot{w}_{\varnothing 2}w_1 = 0$, which form the null-space of Hessian$_+$: $\dot{\boldsymbol{w}}_\varnothing \cdot \boldsymbol{H}_+\dot{\boldsymbol{w}}_\varnothing = 0$. Due to this degeneracy, $\boldsymbol{H}_+$ is non-invertible. Therefore, natural gradient must be computed via the Moore-Penrose pseudo-inverse, which preserves orthogonality to the null-space of Hessian$_+$.

3. Parameter dynamics

In this section, we analyze the learning dynamics of network parameters $\boldsymbol{w}$ under a family of curvature-corrected update rules to discover the shared fundamental property among them.

Steepest gradient descent (SGD) We begin by reproducing the prior analysis on the learning dynamics under gradient descent. SGD update of deep linear networks is given by ($d = 2$ example, η: learning rate)

$$\dot{\boldsymbol{w}} + \eta\boldsymbol{g} = \begin{bmatrix}\dot{w}_1 + \eta\,w_2^\intercal\,\Delta \\ \dot{w}_2 + \eta\,\Delta\,w_1^\intercal\end{bmatrix} = \boldsymbol{0}, \tag{4}$$

which involves nonlinear coupling terms across layers.

Saxe et al. (2013) showed that the complex dynamics of eq (4) can be decomposed into a set of independent *singular mode* dynamics[1], where each singular mode can be seen as a width-1 chain network that consists of 1 neuron and a scalar weight σ_i at each layer. Multiplying all of the scalar weights produces the input-output map of the singular mode

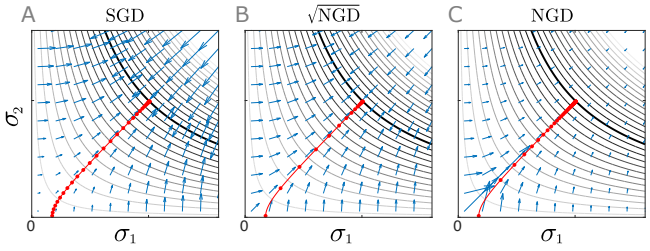

Figure 1. Singular mode dynamics of depth $d = 2$ networks under SGD, $\sqrt{\text{NGD}}$ and NGD. Contour lines visualize constant levels of displacement: $\sigma_\Delta \equiv \sigma_1\sigma_2 - \bar{\sigma}_*$. The optimal solution $\sigma_\Delta = 0$ is shown in black. The vector field visualizes the update vectors normalized by the displacement: $[\dot{\sigma}_1, \dot{\sigma}_2]/|\sigma_\Delta|$. All update dynamics trail hyperbolic paths that conserve $\sigma_1^2 - \sigma_2^2$ (red lines), orthogonal to the null-space of Hessian$_+$ (contour lines), but with different update speed. SGD's update speed linearly scales with Jacobian amplitude $\|j\| = \sqrt{\sigma_1^2 + \sigma_2^2}$ which vanishes for small weights, while NGD's speed scales in a reciprocal manner. In constrast, $\sqrt{\text{NGD}}$'s update speed remains constant with respect to $\|j\|$.

$\bar{\sigma} = \prod_{i=1}^d \sigma_i$, and $\bar{\sigma}_*$ is the corresponding singular value of the desired target map.

A singular mode dynamics of eq (4) is[2]

$$\dot{\sigma}_i + \eta\, \sigma_\Delta\, j_i = 0 \tag{5}$$

where $\sigma_\Delta = \bar{\sigma} - \bar{\sigma}_*$, and $j_i = \prod_{k\neq i} \sigma_k = \bar{\sigma}/\sigma_i$ are the singular values of the displacement and the Jacobian, respectively. This dynamics can be intuitively understood in terms of its path, which trails a hyperbolic curve

$$\sigma_i^2 - \sigma_{i+1}^2 = \text{constant}, \quad \forall i \tag{6}$$

and the update speed along the path

$$\|\dot{\sigma}\| \propto |\sigma_\Delta| \|j\|, \tag{7}$$

where $\|\dot{\sigma}\| \equiv \sqrt{\sum_{i=1}^d \dot{\sigma}_i^2}$, and $\|j\| \equiv \sqrt{\sum_{i=1}^d j_i^2}$. Note that the update speed of SGD vanishes for small Jacobian amplitude $\|j\|$, exhibiting the vanishing speed problem.

Natural gradient descent (NGD) We generalize the above analysis to curvature-corrected learning rules. Natural gradient descent is given by the minimum-norm solution $\min \|\dot{\boldsymbol{w}}\|^2$ that satisfies the constraint

$$\boldsymbol{H}_+ \dot{\boldsymbol{w}} + \eta \boldsymbol{g} = \begin{bmatrix} w_2^\intercal(\dot{\Delta} + \eta\Delta) \\ (\dot{\Delta} + \eta\Delta) w_1^\intercal \end{bmatrix} = \boldsymbol{0}, \tag{8}$$

which yields the Moore-Penrose pseudo-inverse solution

$$\dot{\boldsymbol{w}} + \eta \boldsymbol{J}\Lambda = \begin{bmatrix} \dot{w}_1 + \eta\, w_2^\intercal\, \Lambda \\ \dot{w}_2 + \eta\, \Lambda\, w_1^\intercal \end{bmatrix} = \boldsymbol{0}, \tag{9}$$

(See S.I.). Note that NGD update eq (9) is remarkably similar to the SGD update eq (4) except for the Lagrange multiplier Λ replacing Δ as the normalized displacement. Note that natural gradient retains orthogonality to the null-space of Hessian$_+$: $\dot{\boldsymbol{w}}_\varnothing \cdot \boldsymbol{J}\Lambda = \text{Tr}[\dot{\tilde{w}}^\intercal \Lambda] = 0$.

The singular mode dynamics of eq (9) is

$$\dot{\sigma}_i + \eta\, \sigma_\Lambda\, j_i = 0 \tag{10}$$

where $\sigma_\Lambda = \sigma_\Delta/\|j\|^2$ is the singular value of the normalized displacement. (See S.I.). Note that this dynamics follows the same hyperbolic paths eq (6) as SGD, but with different update speed (See Fig 1C)

$$\|\dot{\sigma}\| \propto \frac{|\sigma_\Delta|}{\|j\|}, \tag{11}$$

which inversely scales with $\|j\|$. Therefore, NGD's update speed explodes for small Jacobian amplitude, reciprocal to SGD's vanishing speed problem.

Fractional Natural Gradient Descent ($\sqrt[q]{\text{NGD}}$) Above results can be generalized to *fractional* curvature-corrected update rules, given by the minimum norm solution to the constraint $\sqrt[q]{\boldsymbol{H}_+}\dot{\boldsymbol{w}} + \eta\boldsymbol{g} = \boldsymbol{0}$, where $\sqrt[q]{\boldsymbol{H}_+}$ denotes the fractional matrix-power of Hessian$_+$ for $q \geq 1$. The solution can be expressed as $\dot{\boldsymbol{w}} + \eta \boldsymbol{J}\Lambda_q = \boldsymbol{0}$, whose singular mode dynamics

$$\dot{\sigma}_i + \eta\, \sigma_\Delta\, j_i\, /\|j\|^{2/q} = 0, \tag{12}$$

interpolates between NGD ($q = 1$) and SGD ($q \to \infty$). This dynamics follows the hyperbolic paths eq (6) with the update speed

$$\|\dot{\sigma}\| \propto |\sigma_\Delta| \|j\|^{1-2/q}. \tag{13}$$

Note that for $q = 2$, termed $\sqrt{\text{NGD}}$, the update speed becomes constant with respect to Jacobian amplitude

$$\|\dot{\sigma}\| = \eta\, |\sigma_\Delta|, \tag{14}$$

which resolves the vanishing/exploding speed problems of SGD and NGD (See Fig 1B).

[1]This decomposition assumes the condition $V_{i+1} = U_i$, $V_1 = V_*$, $U_d = U_*$, between the left and right singular vectors of matrices $\bar{w}_* = U_* A_* V_*^\intercal$, $w_i = U_i A_i V_i^\intercal$, where A_*, A_i's are the singular value matrices with σ_*, σ_i's on the diagonal. This condition is closely satisfied by networks with small initial weights, which exhibit balanced weights ($w_i w_i^\intercal \approx w_{i+1}^\intercal w_{i+1}$), and is widely used for analysis of deep linear networks (Saxe et al., 2013; Lampinen & Ganguli, 2018; Arora et al., 2018a).

[2]Networks with narrow bottleneck layers have singular modes that are *inactive*, which remain frozen without exhibiting any learning dynamics: $\dot{\sigma}_i = 0$. Eq (5),(10) only describe the *active* modes.

3.1. Conservation laws of learning dynamics

The preservation of path shape under curvature correction can be more generally understood in terms of conservation laws. As mentioned in section 2, curvature correction preserves the orthogonality of learning dynamics to the null-space of Hessian$_+$, which originates from the continuous symmetries of deep linear networks. This leads to the following conservation laws.

Theorem 1. *Curvature-corrected update dynamics conserve the following quantities*

$$c_i \equiv w_i w_i^\intercal - w_{i+1}^\intercal w_{i+1} \tag{15}$$

between the adjacent layers of a deep linear network.

Proof. The dot-product between $\dot{\boldsymbol{w}}$ and an arbitrary null transform $\dot{\boldsymbol{w}}_\varnothing$ can be expressed as

$$\dot{\boldsymbol{w}} \cdot \dot{\boldsymbol{w}}_\varnothing = \sum_{i=1}^{d-1} \mathrm{Tr}[(w_i \dot{w}_i^\intercal - \dot{w}_{i+1}^\intercal w_{i+1})\, m_i] = \frac{1}{2}\, \dot{\boldsymbol{c}} \cdot \boldsymbol{m}$$

for arbitrary square matrices m_i. Therefore, orthogonality of curvature-corrected dynamics to the null-space $\dot{\boldsymbol{w}} \cdot \dot{\boldsymbol{w}}_\varnothing = 0, \forall \boldsymbol{m}$ implies the conservation: $\dot{\boldsymbol{c}} = \boldsymbol{0}$. □

Note that the singular mode representation of the conservation law eq (15) yields the hyperbolic paths of eq (6). A restricted case of eq (15) under SGD was shown in Arora et al. (2018b).

This result shows that the path shape of parameter dynamics derives from the intrinsic symmetries of deep network architecture, whereas the temporal profile along the path varies with the specifics of how the update rule handles the curvature.

4. Map dynamics

So far, we focused on the dynamics of weight parameters during learning. In this section, we analyze the dynamics of the input-output map $\dot{\bar{w}} = \boldsymbol{J}^\intercal \dot{\boldsymbol{w}}$, whose singular-mode representation is

$$\dot{\bar{\sigma}} = \sum_{i=1}^{d} \dot{\sigma}_i\, j_i = -\eta\, (\bar{\sigma} - \bar{\sigma}_*)\, \|j\|^{2(1-1/q)}, \tag{16}$$

where the $\sqrt[q]{\mathrm{NGD}}$ update eq (12) is considered. Previously, Saxe et al. (2013) analyzed the effect of depth on map dynamics under SGD update, which is generalized here to include the effect of curvature correction.

Note that the effect of curvature is entirely contained in the Jacobian amplitude term, where as the displacement term drives the map dynamics toward the target map strength $\bar{\sigma}_*$.

Although the Jacobian amplitude term depends on individual layer weights σ_i, the main characteristics of how it scales with the overall weight strength can be concisely captured by considering the balanced condition $c_i = 0$, in which the layer weights share the value $\sigma_i = \bar{\sigma}^{1/d}$[3]. In this condition, eq (16) reduces to

$$\dot{\bar{\sigma}} = -\bar{\eta}\, (\bar{\sigma} - \bar{\sigma}_*)\, \bar{\sigma}^p \quad \left(p \equiv \frac{2(d-1)(q-1)}{d\,q}\right) \tag{17}$$

where $\bar{\eta} \equiv \eta\, d^{1-1/q}$ is the *depth-calibrated* learning rate. The exponent p indicates the combined effect of depth and curvature correction on the nonlinearity of map dynamics (See Table 1). Figure 2 shows the following notable closed-form solutions, as well as the $p = 2$ case:

$$\begin{aligned}
\bar{\sigma}_{(t)} &= \bar{\sigma}_*(1 - e^{-\bar{\eta} t}) && (p = 0)\\
\bar{\sigma}_{(t)} &= \bar{\sigma}_* \tanh^2(\bar{\eta}\sqrt{\bar{\sigma}_*}\, t/2) && (p = 1/2)\\
\bar{\sigma}_{(t)} &= \frac{\bar{\sigma}_*}{1 + (\bar{\sigma}_*/\bar{\sigma}_{(0)} - 1) e^{-\bar{\eta}\bar{\sigma}_* t}} && (p = 1)
\end{aligned}$$

where zero initial condition $\bar{\sigma}_{(0)} = 0$ is assumed for $p < 1$.

4.1. Shallow networks ($d = 1$, $p = 0$)

Shallow networks exhibit linear map dynamics for all levels of curvature correction. Also, it exhibits a constant convergence rate $\bar{\eta}$ for all singular modes regardless of their target map strength $\bar{\sigma}_*$.

4.2. Deep networks ($d \geq 2$)

NGD ($q = 1, p = 0$) NGD continues to exhibit linear map dynamics for deep networks and constant time-scale of learning $1/\bar{\eta}$ for all singular modes regardless of the target map strength $\bar{\sigma}_*$. Note that the exploding parameter update speed is necessary to sustain the finite map convergence rate at saddle-points where gradient vanishes. Thus, NGD's smooth map dynamics entails sacrificing the stability of parameter dynamics.

SGD ($q \to \infty,\ p = 2 - 2/d$) SGD exhibits highly nonlinear map dynamics with the exponent p ranging from

[3] The balanced condition is closely satisfied by networks trained from small initial weights. See section 7.

Table 1. Exponent p for given depth d and curvature correction q

p	$q = 1$	$q = 2$	$\cdots$	$q \to \infty$
$d = 1$	0	0	$\cdots$	0
$d = 2$	0	1/2		1
$\vdots$	$\vdots$		$\ddots$	
$d \to \infty$	0	1		2

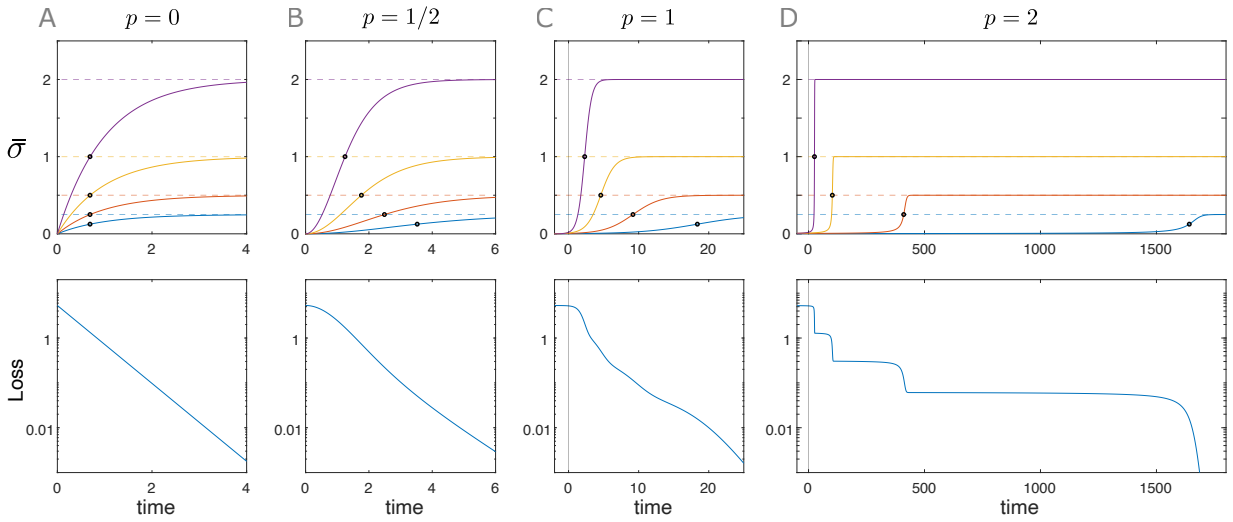

Figure 2. Learning dynamics eq (17) of the input-output map under different exponents p. Top: Learning curves of the map strength in singular mode representation: $\bar{\sigma}_{(t)}$. Dashed lines show the target map strengths $\bar{\sigma}_*$. Half-max points (black circles) are shown to visualize the time-scale of learning, which scales with the target map strength as $\bar{\sigma}_*^{-p}$. Bottom: Corresponding loss profiles. $\bar{\eta} = 1$. Initial conditions: $\bar{\sigma}_{(0)} = 0$ for $p < 1$, $\bar{\sigma}_{(0)} = \bar{\sigma}_*/100$ for $p \geq 1$.

$p = 1$ for one-hidden-layer networks to $p \to 2$ in infinite depth limit. Due to its vanishing speed problem near the saddle-point, SGD's escape-time from zero initial condition ($\sigma_{(0)} \to 0$) diverges as $\mathcal{O}(-\log \bar{\sigma}_{(0)})$ for one-hidden-layer networks and as $\mathcal{O}(1/\bar{\sigma}_{(0)}^{p-1})$ for deeper networks. This results in the characteristic, sigmoidal-shaped learning curves (Fig 2 C,D). Moreover, the time-scale of learning scales as $1/\bar{\eta}\bar{\sigma}_*^p$, such that the singular modes with stronger targets $\bar{\sigma}_*$ learn faster than the singular modes with weaker targets. The combination of sigmoidal learning curves and wide separation of time-scales produces the characteristic loss profiles of deep learning that exhibit long plateaus followed by rapid transitions (Fig 2 D).

$\sqrt{\text{NGD}}$ ($q = 2,\ p = 1 - 1/d$) $\sqrt{\text{NGD}}$ exhibits moderately nonlinear map dynamics with the exponent p ranging from $p = 1/2$ for one-hidden-layer networks (Fig 2B) up to $p \to 1$ in infinite depth limit (Fig 2C). Due to its non-vanishing update speed, $\sqrt{\text{NGD}}$ escapes from the saddle-point within finite time for all depth, exhibiting polynomially growing learning curves near zero $\bar{\sigma}(t) \propto t^{1/(1-p)}$, followed by gradual transitions to convergence. $\sqrt{\text{NGD}}$ exhibits differential learning time-scale across singular modes proportional to $1/\bar{\eta}\bar{\sigma}_*^p$, although at milder levels than SGD's.

4.3. Effective depth

Equation (17) shows that depth and curvature correction use the same mechanism for adding and lessening the nonlinearity of map dynamics. Therefore, the effect of curvature correction can be intuitively understood as reducing the *effective depth* of the network,

$$d_{\text{eff}} = \frac{dq}{d + q - 1} \tag{18}$$

defined as the depth that yields the same degree of nonlinearity in the absence of curvature correction. The effective depth approaches the actual depth $d_{\text{eff}} \to d$ in the SGD limit $q \to \infty$, and similarly, it approaches $d_{\text{eff}} \to q$ in the infinite depth limit $d \to \infty$. Thus, d_{eff} is upper-bounded by q. For $\sqrt{\text{NGD}}$, the nonlinearity of map learning dynamics is always less than that of one-hidden-layer networks: $d_{\text{eff}} < 2$.

5. Effect of block-diagonal approximations

Block-diagonal NGD (NGD-d) Due to the computational cost of numerically estimating and inverting Hessian$_+$, full implementation of NGD does not scale well to practical problems in deep learning applications. Instead, most second-order methods use *layer-restricted*, or *block-diagonal* approximations of Hessian$_+$ (Martens & Grosse, 2015; Ba et al., 2016; Grosse & Martens, 2016; Martens et al., 2018; Bernacchia et al., 2018) that separately apply curvature corrections for each layer while ignoring the curvature relationship between layers (*i.e.* the off-block-diagonal terms): *i.e.*

$$\dot{w}_i + \eta H_i^{-1} g_i / d = 0, \tag{19}$$

called NGD-d, whose singular mode dynamics is

$$\dot{\sigma}_i + \frac{\eta\, \sigma_\Delta}{d\, j_i} = 0, \tag{20}$$

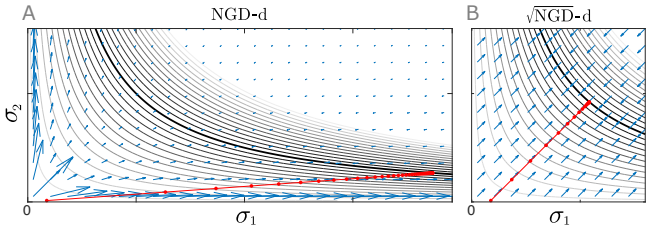

Figure 3. Same as Figure 1, but under block-diagonally approximated curvature corrections. The vector fields are no longer orthogonal to the contour lines. (A) NGD-d exhibits a radially diverging vector field that conserves σ_1/σ_2. Note that its trajectory traverses the contour lines at the same rate as NGD in Fig 1C (red dots). (B) $\sqrt{\text{NGD}}$-d exhibits a parallel vector field of constant direction and amplitude that conserves $|\sigma_1| - |\sigma_2|$.

where NGD's Jacobian amplitude normalization factor $\|j\|^2$ in eq (10) is substituted by the layer-restricted factor j_i^2.

Note that NGD-d exhibits the same linear map dynamics as NGD: $\dot{\bar{\sigma}} = \sum_{i=1}^{d} \dot{\sigma}_i \, j_i = -\eta \, (\bar{\sigma} - \bar{\sigma}_*)$, which was also found in Bernacchia et al. (2018).

In parameter dynamics, however, block-diagonal approximation significantly modifies the path shape to be radially diverging to conserve σ_i/σ_{i+1} as constants of motion (Fig 3A). It also further increases the update speed $\|\dot{\sigma}\|^2_{\text{NGD-d}} \geq \|\dot{\sigma}\|^2_{\text{NGD}}$, since $\frac{1}{d}\sum_{i=1}^{d}(j_i^2)^{-1} \geq (\frac{1}{d}\sum_{i=1}^{d} j_i^2)^{-1}$ (Jensen's inequality). This difference is due to the significant non-zero components in the null-space of Hessian$_+$ that do not contribute to the network performance. Consequently, block-diagonal NGD converges to less efficient, large norm solutions that are highly sensitive to initial conditions and noise during training (Fig 3A, red line).

Block-diagonal $\sqrt{\text{NGD}}$ ($\sqrt{\text{NGD}}$-d) Block-diagonal approximation of $\sqrt{\text{NGD}}$ causes much milder modification of parameter dynamics, whose singular mode dynamics

$$\dot{\sigma}_i + \frac{\eta \, \sigma_\Delta}{\sqrt{d}} \, \text{sign}(j_i) = 0, \tag{21}$$

generates straight parallel paths that conserve the absolute difference of singular values across layers $|\sigma_i| - |\sigma_{i+1}|$ as constants of motion[4] (Fig 3B). These non-diverging paths yield stable parameter dynamics that converge to close solutions to SGD's solutions. Moreover, $\sqrt{\text{NGD}}$-d retains the non-vanishing/exploding update speed of $\sqrt{\text{NGD}}$: $\|\dot{\sigma}\| = \eta \, |\sigma_\Delta|$.

[4]More generally, $\sqrt[q]{\text{NGD}}$-d conserves $\sigma_i^{2(1-1/q)} - \sigma_k^{2(1-1/q)}$ as constants of motion.

6. Numerical simulations

To test the main theoretical results, we conducted a simple synthetic data experiment, in which the training and the testing datasets are generated from a random teacher network as $y^\mu = \bar{w}_{\text{teacher}} x^\mu + z^\mu$, where $x^\mu \in \mathbb{R}^N$ is the whitened input data, $y^\mu \in \mathbb{R}^N$ is the output, $z^\mu \in \mathbb{R}^N$ is the noise (Lampinen & Ganguli, 2018). The input-output map of the teacher network $\bar{w}_{\text{teacher}} \in \mathbb{R}^{N\times N}$ has a low-rank structure (rank 3, Fig 4A) and the student is a depth $d = 4$ linear network of constant width $N = 16$. The number of training dataset $\{x^\mu, y^\mu\}_{\mu=1}^{P}$ is set to be $P = N$, which makes the learning problem most susceptible to overfitting. The student network is trained from small random initial weights.

Hessian$_+$ blocks are computed as described in Bernacchia et al. (2018); Botev et al. (2017) and combined to obtain full Hessian$_+$. NGD-d and $\sqrt{\text{NGD}}$-d only used the diagonal blocks. Numerical pseudo-inverses (and sqrt-inverses) are computed via singular value decomposition (SVD). For numerical stability, NGD and NGD-d used Levenberg-Marquardt damping of $\epsilon = 10^{-5}$ and update-speed clipping. $\sqrt{\text{NGD}}$ and $\sqrt{\text{NGD}}$-d did not require such corrections.

Fig 5A,B show the learning trajectories of weight parameters, which reflect the mixed dynamics of multiple singular modes. Despite the seemingly different trajectories, SGD and its curvature-corrected update rules (NGD, $\sqrt{\text{NGD}}$) all conserve the same quantities of eq (15), as confirmed in Figure 5C, indicating the same paths followed by their underlying singular mode components. As a result, the parameter dynamics under SGD, NGD, and $\sqrt{\text{NGD}}$ converge to the same solution. Moreover, due to its non-diverging dynamics, $\sqrt{\text{NGD}}$-d's parameter dynamics stays close to $\sqrt{\text{NGD}}$ dynamics, even though it does not obey the same conservation law. In contrast, NGD-d's parameter dynamics quickly diverges from NGD dynamics, and tends to converge to solutions with much larger parameter values. It is also extremely sensitive to small differences in learning conditions such as learning rate and clipping value.

Figure 4D shows the learning trajectories of the input-output map, which closely matches the analytical predictions of section 4, except that NGD and NGD-d no longer exhibit exponential convergence due to the finite clipping of update-speed. Due to differential time-scales of learning, SGD preferentially learns the strong singular modes first, *i.e.* the signal dimensions, before overfitting the small noise dimensions. $\sqrt{\text{NGD}}$ and $\sqrt{\text{NGD}}$-d also exhibit differential learning time-scales across singular modes, and thus achieve low generalization error via early-stopping (Fig 4C). In contrast, NGD and NGD-d learn all singular modes at the same time, and therefore overfit the noise dimensions from the very beginning, which harms the generalization performance.

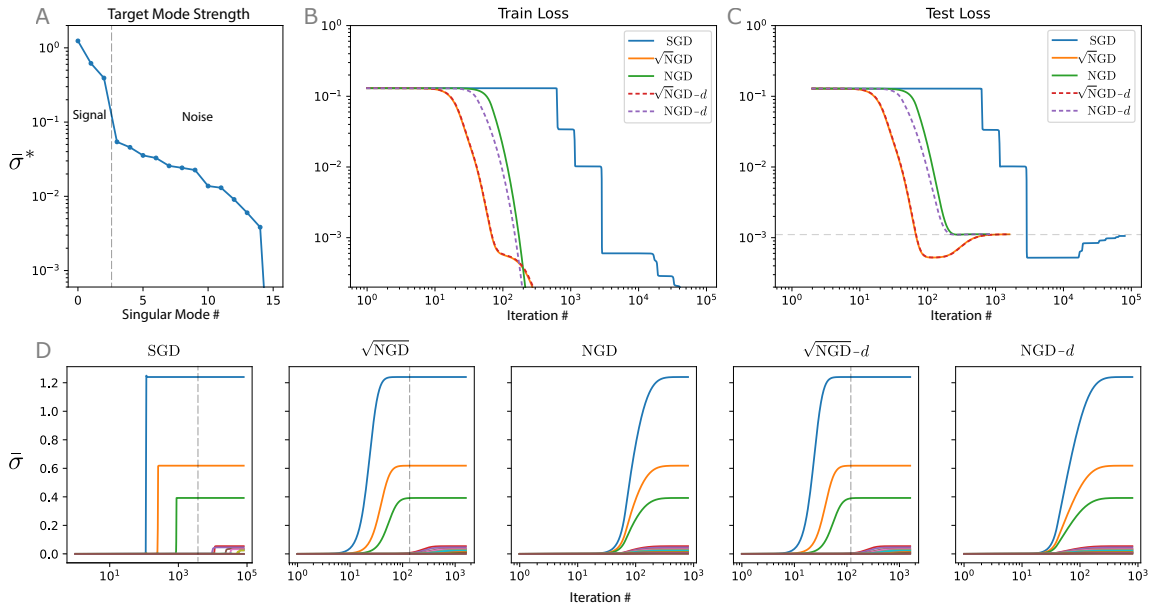

Figure 4. Learing dynamics of the input-output map in teacher-student task. (A) Singular values of the target map $\bar{\sigma}^*$ in training dataset. (B) Train-loss profile. (C) Test-loss profile. (D) Singular mode dynamics of the input-output map (Similar to Fig 2). SGD, $\sqrt{\text{NGD}}$, $\sqrt{\text{NGD}}$-d learn the signal dimensions before the noise dimensions, which allows achieving low generalization error via early-stopping (vertical dashed lines). NGD and NGD-d learn the noise dimensions from very beginning, simultaneously with the signal dimensions.

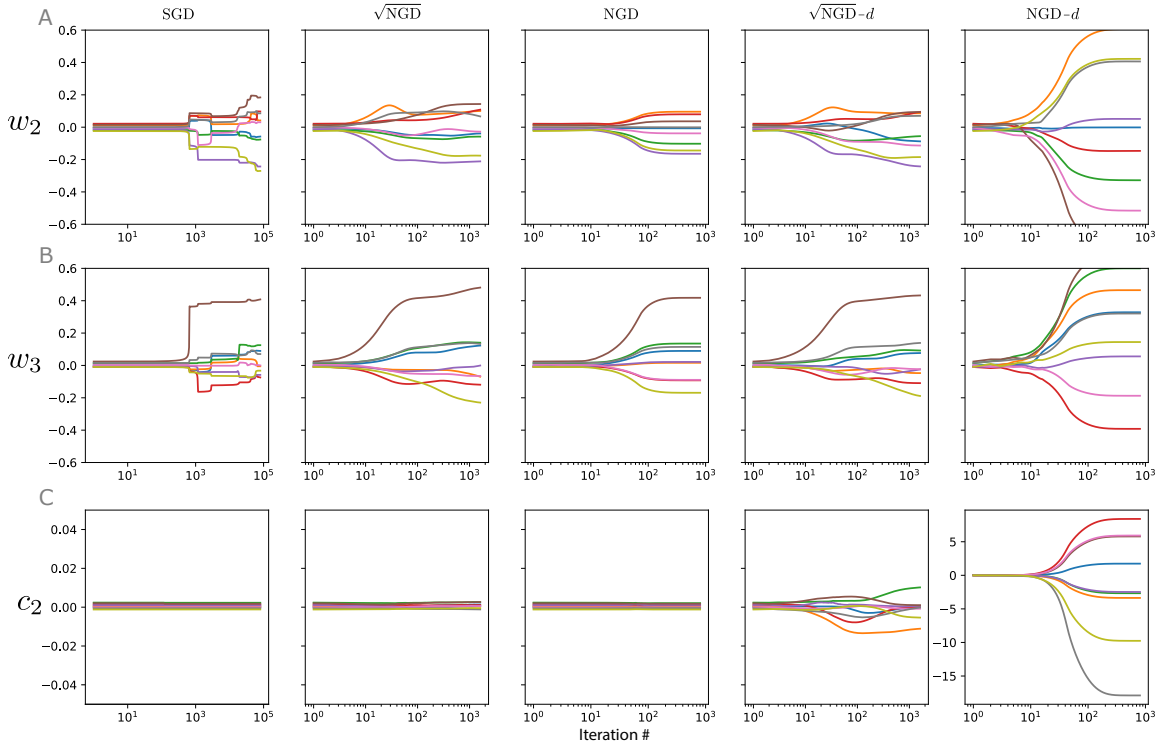

Figure 5. Parameter dynamics during the time-course of learning. Only a subset of the weight matrix elements are plotted. (A,B) Weight matrices of layer 2 and 3: w_2, w_3. (C) Squared-difference of the weights eq (15): $c_2 \equiv w_2 w_2^{\intercal} - w_3^{\intercal} w_3$. Due to small initial weights, the networks are initialized at nearly balanced condition: $c_2(0) \approx 0$. SGD, $\sqrt{\text{NGD}}$, and NGD updates keeps this difference small (except for numerical deviation due to finite update step size), which maintains weight balance across layers. $\sqrt{\text{NGD}}$-d maintains weight balance by conserving a different quantity. (See text.) NGD-d does not maintain weight balance.

7. Discussion

Our analysis shows that curvature correction preserves the path of parameter dynamics, while modifying the temporal profile of map dynamics to reduce the nonlinear effect of depth. This mechanism has important implications for the stability and generalization properties of second-order learning algorithms.

Parameter vs Map dynamics Curvature correction accelerates convergence by reducing the nonlinearity of learning dynamics caused by network depth. However, this process involves a trade-off between map dynamics and parameter dynamics: While NGD's full curvature correction completely remove the nonlinearity of input-output map's dynamics, it risks the stability of parameter dynamics, which explodes at saddle-points. In contrast, $\sqrt{\text{NGD}}$'s fractional curvature correction yields stable parameter dynamic by eliminating the vanishing/exploding speed problems, while exhibiting significantly reduced level of nonlinearity in map dynamics.

Implicit bias for regularization SGD exhibits strong implicit bias for efficiently extracting the low-rank statistics in datasets, such as finding matrix factorizations with minimum nuclear norm (Gunasekar et al., 2017; Arora et al., 2019), as well as avoiding overfitting in deep networks by early stopping (Advani & Saxe, 2017; Lampinen & Ganguli, 2018), which is crucial for achieving good generalization performance. This is due to SGD's preference to learning the stronger singular modes of dataset faster than weaker modes (Saxe et al., 2013). Our analysis reveals that this implicit bias can be affected by curvature correction. Especially, NGD is prone to overfitting by simultaneously learning both the signal and the noise dimensions of data. This explains the recent observation that second-order optimization often leads to worse generalization performance (Zhang et al., 2018). In contrast, $\sqrt{\text{NGD}}$ retains the differential time-scales of learning and achieves good generalization performance.

Weight balance Another implicit regularization property of SGD is in maintaining the weight balance across layers, which is essential for the stability of learning dynamics and for convergence analysis (Du et al., 2018). We showed that this property extends to all curvature corrected learning dynamics as a direct consequence of the conservation law eq (15) on path shape. However, the widely-used block-diagonal approximation of NGD (e.g. K-FAC) (Ba et al., 2016; Grosse & Martens, 2016; Martens et al., 2018) causes the path shape to diverge, and thereby breaks the conservation law. Consequently, under NGD-d, weight parameterization can potentially diverge to unbounded solutions, even when trained from small initial weights. In contrast, $\sqrt{\text{NGD}}$-d exhibits non-diverging path shape that maintains balance across layers.

Our analysis provides deep theoretical insights for the effect of curvature on the learning process of deep linear neural networks, as well as implications for designing more robust second-order algorithms for practical applications. Further analysis is needed for networks with nonlinear activation functions,

References

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

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

Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018a.

Arora, S., Cohen, N., and Hazan, E. On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*, 2018b.

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization. *arXiv preprint arXiv:1905.13655*, 2019.

Ba, J., Grosse, R., and Martens, J. Distributed second-order optimization using kronecker-factored approximations. 2016.

Bartlett, P. L., Helmbold, D. P., and Long, P. M. Gradient descent with identity initialization efficiently learns positive-definite linear transformations by deep residual networks. *Neural computation*, 31(3):477–502, 2019.

Bernacchia, A., Lengyel, M., and Hennequin, G. Exact natural gradient in deep linear networks and its application to the nonlinear case. In *Advances in Neural Information Processing Systems*, pp. 5941–5950, 2018.

Botev, A., Ritter, H., and Barber, D. Practical gauss-newton optimisation for deep learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 557–565. JMLR. org, 2017.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pp. 2933–2941, 2014.

Du, S. S. and Hu, W. Width provably matters in optimization for deep linear neural networks. *arXiv preprint arXiv:1901.08572*, 2019.

Du, S. S., Hu, W., and Lee, J. D. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, pp. 384–395, 2018.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Grosse, R. and Martens, J. A kronecker-factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning*, pp. 573–582, 2016.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 6151–6159, 2017.

Heskes, T. On "natural" learning and pruning in multilayered perceptrons. *Neural Computation*, 12(4):881–901, 2000.

Lampinen, A. K. and Ganguli, S. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *arXiv preprint arXiv:1809.10374*, 2018.

Martens, J. Deep learning via hessian-free optimization. In *ICML*, volume 27, pp. 735–742, 2010.

Martens, J. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

Martens, J. and Grosse, R. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417, 2015.

Martens, J., Ba, J., and Johnson, M. Kronecker-factored curvature approximations for recurrent neural networks. 2018.

Osawa, K., Tsuji, Y., Ueno, Y., Naruse, A., Yokota, R., and Matsuoka, S. Large-scale distributed second-order optimization using kronecker-factored approximate curvature for deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12359–12367, 2019.

Pascanu, R. and Bengio, Y. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.

Poggio, T., Liao, Q., Miranda, B., Banburski, A., Boix, X., and Hidary, J. Theory iiib: Generalization in deep networks. *arXiv preprint arXiv:1806.11379*, 2018.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

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

Supplemental Materials

S.I.1. Moore-Penrose inverse solution: eq (9) in Section 3.2

In section 4, we find the Moore-Penrose inverse solution that minimizes the update norm $\dot{\boldsymbol{w}} \cdot \dot{\boldsymbol{w}} = \sum_i \mathrm{Tr}[\dot{w}_i \dot{w}_i^\intercal]$ while satisfying the natural gradient constraint: ($d = 2$ example)

$$\boldsymbol{H}\dot{\boldsymbol{w}} + \eta \boldsymbol{g} = \begin{bmatrix} w_2^\intercal(\dot{\Delta} + \eta\Delta) \\ (\dot{\Delta} + \eta\Delta) w_1^\intercal \end{bmatrix} = \boldsymbol{0}. \tag{S.I.1}$$

This constrained optimization problem is described by the following Lagrangian:

$$\begin{aligned} \mathcal{L}(\dot{w}_1, \dot{w}_2, \Lambda_2, \Lambda_2) &= (\dot{w}_1 \cdot \dot{w}_1 + \dot{w}_2 \cdot \dot{w}_2)/2 \\ &+ \Lambda_1 \cdot w_2^\intercal(\dot{\Delta} + \eta\Delta) + \Lambda_2 \cdot (\dot{\Delta} + \eta\Delta) w_1^\intercal, \end{aligned} \tag{S.I.2}$$

where $\dot{\Delta} = w_2 \dot{w}_1 + \dot{w}_2 w_1$, and dot notation denotes inner-product: $a \cdot b \equiv \mathrm{Tr}[a^\intercal b]$. Optimality condition on $\dot{w}_i$ yields

$$\partial\mathcal{L}/\partial\dot{w}_1 = \dot{w}_1 + w_2^\intercal(w_2\Lambda_1 + \Lambda_2 w_1) = 0 \tag{S.I.3}$$

$$\partial\mathcal{L}/\partial\dot{w}_2 = \dot{w}_2 + (w_2\Lambda_1 + \Lambda_2 w_1) w_1^\intercal = 0 \tag{S.I.4}$$

which, via change of variables $\Lambda \equiv (w_2\Lambda_1 + \Lambda_2 w_1)/\eta$, reduces to

$$\dot{w}_1 + \eta w_2^\intercal \Lambda = 0 \tag{S.I.5}$$

$$\dot{w}_2 + \eta \Lambda w_1^\intercal = 0 \tag{S.I.6}$$

which can be plugged into the optimality condition on Λ_i

$$\partial\mathcal{L}/\partial\Lambda_1 = w_2^\intercal(\dot{\Delta} + \eta\Delta) = 0 \tag{S.I.7}$$

$$\partial\mathcal{L}/\partial\Lambda_2 = (\dot{\Delta} + \eta\Delta) w_1^\intercal = 0 \tag{S.I.8}$$

to produce a linear equation for Λ_i:

$$w_2^\intercal S(\Lambda) = S(\Lambda) w_1^\intercal = 0 \tag{S.I.9}$$

$$\text{where} \quad S(\Lambda) = (w_2 w_2^\intercal)\Lambda + \Lambda(w_1^\intercal w_1) - \Delta. \tag{S.I.10}$$

Note that if w_2, w_1 are invertible, it is easy to see that eq (S.I.7)(S.I.8) reduce to exponentially converging dynamics $\dot{\Delta} + \eta\Delta = \dot{\bar{w}} + \eta(\bar{w} - \bar{w}^*) = 0$, with the solution of $S(\Lambda) = 0$ driving the parameter update eq (S.I.5)(S.I.6). This result also holds true for the over-complete cases, where the hidden layer width is larger than the minimum of input layer or output layer size. For the under-complete cases, *i.e.* with bottleneck hidden layers, the exponential convergence applies only to the subspace dimensions permitted by the bottleneck, with the other dimensions remain frozen.

S.I.2. Singular mode analysis eq (10) in Section 3.2

We follow the SVD-based analysis of Saxe et al. (2013) under the aligned singular vector condition. We introduce $\sigma_i, \bar{\sigma}, \sigma_\Delta, \sigma_\Lambda, \sigma_S$ that represent the singular values of $w_i, \bar{w}, \Delta, \Lambda, S(\Lambda)$ of one singular mode, and $j_i = \prod_{k\neq i}\sigma_k = \bar{\sigma}/\sigma_i$. In this representation, eq (S.I.5)(S.I.6) reduce to

$$\dot{\sigma}_i = -\eta\sigma_\Lambda j_i \tag{S.I.11}$$

whereas, eq (S.I.9)(S.I.10) reduce to

$$\sigma_i\sigma_S = 0 \qquad \forall i \tag{S.I.12}$$

$$\sigma_S = \sum_{i=1}^{d} j_i^2\sigma_\Lambda - \sigma_\Delta \tag{S.I.13}$$

Therefore, an *active* singular mode with non-zero σ_i's must have $\sigma_S = 0$, which results in

$$\sigma_\Lambda = \frac{\sigma_\Delta}{\sum_{i=1}^{d} j_i^2} \tag{S.I.14}$$

which produces the result of the main text.

Relation to Regularized NGD Alternative interpolation solves $(\boldsymbol{H}_+\dot{\boldsymbol{w}} + \eta\boldsymbol{g}) + \epsilon\boldsymbol{I}(\dot{\boldsymbol{w}} + \eta\boldsymbol{g}) = \boldsymbol{0}$ ($\epsilon \geq 0$), which yields the regularized (or damped) inverse

$$\dot{\boldsymbol{w}} = -\eta(\epsilon+1)(\epsilon\boldsymbol{I} + \boldsymbol{H}_+)^{-1}\boldsymbol{g},^{5} \tag{S.I.15}$$

similar to Levenberg-Marquardt damping (less the $(\epsilon+1)$ term), whose singular mode dynamics is

$$\dot{\sigma}_i + \eta\,\sigma_\Delta\,\frac{j_i}{\|j\|}\left(\frac{a\|j\|+1}{a+\|j\|}\right) = 0, \quad (a \equiv \epsilon/\|j\|) \tag{S.I.16}$$

where the ratio $a \equiv \epsilon/\|j\|$ describes the effective degree of interpolation between NGD ($a \to 0$) and SGD ($a \to \infty$). Note that a should be large enough to provide sufficient damping, but not too large to nullify the effect of curvature correction, which is difficult to simultaneously satisfy across all singular modes with fixed ϵ. $\sqrt{\text{NGD}}$ can be considered as providing adaptively tuned regularization ($a = 1$) for all singular modes, where the regularization is most effective.

S.I.3. Nonlinear network training on MNIST classification task

Here, we experimented with the effect of curvature corrections in non-linear networks by training a 5 layer network for MNIST classification task. Network of layer size [784,300,100,30,10] with alternation between dense layer and ReLU layer were used. The weights were initialized as orthogonal matrices, as suggested in (Saxe et al., 2013), with various gains that range between 1 and 10^{-6}, which translates to the initial singular value of $\bar{\sigma}_o$ ranging between 1 and 10^{-24}. batch-size of 128. As predicted, training with SGD and NGD-d works well for $\sigma_o = 1$, but they struggle for smaller initial gains. For the NGD-d update rule, the learning rate and the Levenberg-Marquardt damping term must be adjusted to scale with the gain for numerical stability. However, this leads to reduced effect of curvature-correction. In contrast, $\sqrt{\text{NGD}}$-d successfully trains the network using a fixed learning rate, without introducing any damping term for the inverse.

[5] This expression reduces to SGD in the limit $\epsilon \to \infty$, which differs from the usual regularized inverse $\dot{\boldsymbol{w}} = -\eta(\epsilon\boldsymbol{I} + \boldsymbol{H}_+)^{-1}\boldsymbol{g}$, which reduces to $\boldsymbol{0}$.

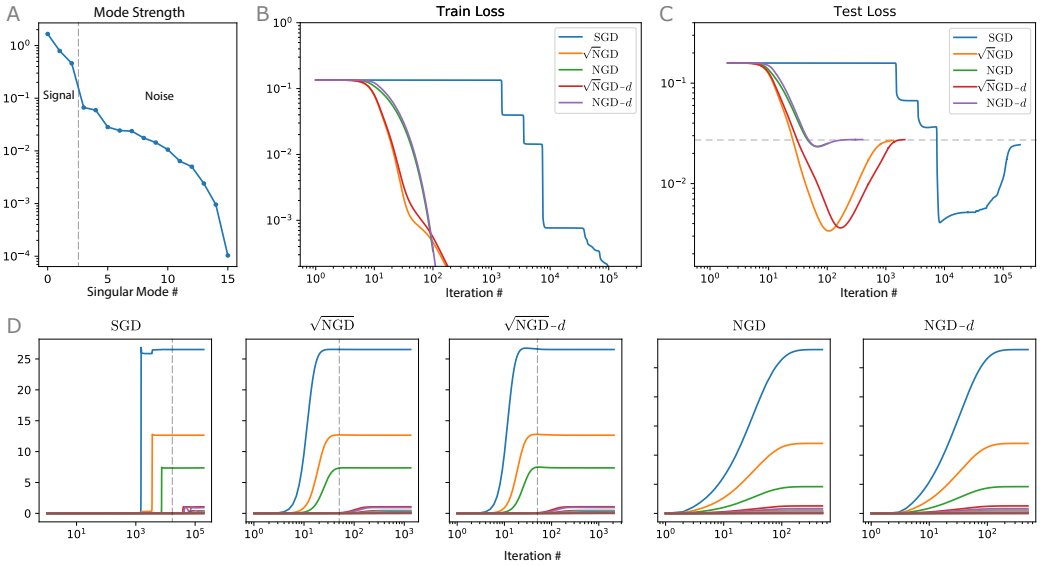

Figure S.I.1. Learing dynamics of the input-output map in teacher-student task with non-whitened input data, exhibiting pronounced generalization error in the test-loss

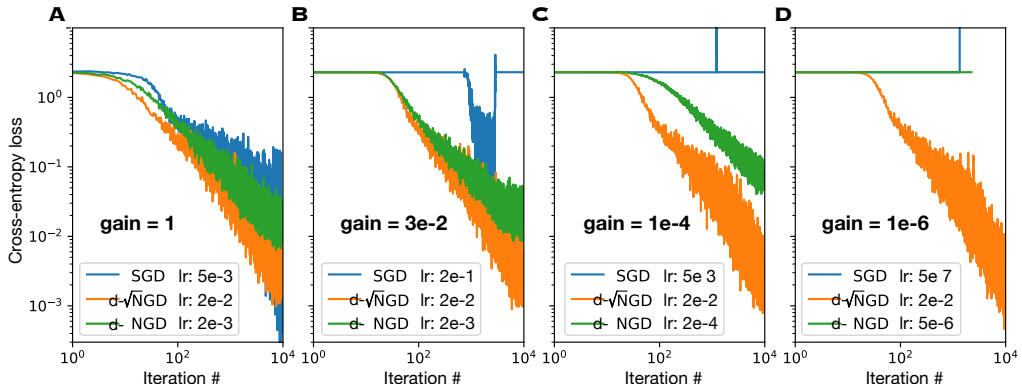

Figure S.I.2. Reviving vanishing gradient: Training a 5-layer ReLU network on MNIST dataset. Weights are initialized to be orthonormal matrices with various gains that range between 1 and 10^{-6}. NGD-d requires a small damping term for inverting Hessian $(\epsilon \boldsymbol{I} + \boldsymbol{H}_+)^{-1}$ with $\epsilon = [10^{-3}, 10^{-3}, 10^{-6}, 10^{-7}]$ for numerical stability. $\sqrt{\text{NGD}}$-d requires no such damping. batch-size = 128. Network architecture: [784,300,100,30,10].