EXACT ANALYSIS OF CURVATURE CORRECTED LEARNING DYNAMICS IN DEEP LINEAR NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks exhibit complex learning dynamics due to the highly non-convex loss landscape, which causes slow convergence and vanishing gradient problems. Second order approaches, such as natural gradient descent, mitigate such problems by neutralizing the effect of potentially ill-conditioned curvature on the gradient-based updates, yet precise theoretical understanding on how such curvature correction affects the learning dynamics of deep networks has been lacking. Here, we analyze the dynamics of training deep neural networks under a generalized family of natural gradient methods that applies curvature corrections, and derive precise analytical solutions. Our analysis reveals that curvature corrected update rules preserve many features of gradient descent, such that the learning trajectory of each singular mode in natural gradient descent follows precisely the same path as gradient descent, while only accelerating the temporal dynamics along the path. We also show that layer-restricted approximations of natural gradient, which are widely used in most second order methods (*e.g.* K-FAC), can significantly distort the learning trajectory into highly diverging dynamics that significantly differs from true natural gradient, which may lead to undesirable network properties. We also introduce fractional natural gradient that applies partial curvature correction, and show that it provides most of the benefit of full curvature correction in terms of convergence speed, with additional benefit of superior numerical stability and neutralizing vanishing/exploding gradient problems, which holds true also in layer-restricted approximations.

1 INTRODUCTION

Despite recent advances in deep learning, training deep neural networks is still non-trivial and time-consuming process, which often exhibits alternating periods of fast learning and plateau phases where learning mostly stalls. Such characteristic learning profiles arise from the fact that the loss surface of deep neural networks is highly non-convex due to the prevalence of saddle-points and poorly-conditioned curvature (Dauphin et al., 2014), where gradient-based optimization methods perform poorly. Second order optimization methods, such as natural gradient descent (NGD), have been proposed to mitigate these problems by compensating for the negative effect of saddle-points and poorly-conditioned curvature in learning dynamics. However, theoretical analysis on the deep learning dynamics under such curvature correction has been lacking.

Recent works have shown that the effect of curvature of loss landscape on deep learning dynamics can be well captured by deep linear networks (Saxe et al., 2013). Here, we present analytical solutions to the curvature corrected learning dynamics in deep linear networks to gain critical understanding on the advantage and effects of applying curvature corrections.

2 PROBLEM SETUP

Consider a linear multi-layer network of depth d that consists of an input layer, $d-1$ hidden layers, an output layer, and weight matrices $\boldsymbol{w} \equiv \{w_i\}_{i=1}^{d}$ that connect adjacent layers. This network learns

the input-output statistics of a dataset by minimizing the squared-error loss:

$$L(\boldsymbol{w}) = \frac{1}{2}\mathbb{E}[\|\bar{w}x - y\|^2] = \mathrm{Tr}\left[\frac{1}{2}(\bar{w} - w_*)\Sigma_x(\bar{w} - w_*)^\intercal\right] + constant, \tag{1}$$

where $\bar{w} \equiv \prod_{i=1}^{d} w_i = w_d \cdots w_1$ is the input-output map of the network, $\mathbb{E}[\cdot]$ is the full-batch expectation over the dataset $\{x^\mu, y^\mu\}_\mu$, $\Sigma_x \equiv \mathbb{E}[xx^\intercal]$ is the input correlations, and $w_* \equiv \Sigma_{yx}\Sigma_x^{-1} \equiv \mathbb{E}[yx^\intercal]\,\Sigma_x^{-1}$ is the optimum for $\bar{w}$. Eq (1) can be expressed as $L(\boldsymbol{w}) = \mathrm{Tr}\left[\frac{1}{2}\Delta\Sigma_x\Delta^\intercal\right] + constant$, where $\Delta \equiv \bar{w} - w_*$ is the displacement from the optimum. For the ease of exposition, the input correlation is assumed to be whitened $\Sigma_x = I$, although it is not essential to the analysis.

Gradient and Hessian We introduce bold symbols to collectively represent the derivatives of network parameters in array form: For example, $\dot{\boldsymbol{w}} \equiv \begin{bmatrix}\dot{w}_1 \\ \dot{w}_2\end{bmatrix}$ and $\boldsymbol{g} \equiv \begin{bmatrix}\frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2}\end{bmatrix} = \begin{bmatrix}w_2^\intercal\,\Delta \\ \Delta\, w_1^\intercal\end{bmatrix}$ represent the continuous-time weight update and the gradient of a depth $d = 2$ network, respectively. Hessian is fully characterized by its operation on weight update, which, by definition, produces gradient update:

$$\boldsymbol{H}_{\text{true}}\dot{\boldsymbol{w}} = \dot{\boldsymbol{g}} = \begin{bmatrix}w_2^\intercal\dot{\Delta} + \dot{w}_2^\intercal\Delta \\ \dot{\Delta}w_1^\intercal + \Delta\dot{w}_1^\intercal\end{bmatrix}. \qquad (\dot{\Delta} = \dot{\bar{w}} = w_2\dot{w}_1 + \dot{w}_2 w_1) \tag{2}$$

However, true Hessian-based methods (*e.g.* Newton-Raphson method) can converge to any extrema types including saddle-points. To guarantee convergence to local minima, natural gradient methods use positive semi-definite (PSD) portion/approximations of Hessian (*e.g.* Fisher matrix (Amari, 1998; Heskes, 2000; Martens & Grosse, 2015; Bernacchia et al., 2018), Gauss-Newton matrix (Martens, 2014; Botev et al., 2017; Roux et al., 2008)), which corresponds to discarding the undifferentiated Δ terms from eq (2):

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

This operation is indeed PSD, since $\dot{\boldsymbol{w}} \cdot \boldsymbol{H}_+\dot{\boldsymbol{w}} = \mathrm{Tr}[\dot{w}_1^\intercal w_2^\intercal\dot{\Delta} + \dot{w}_2^\intercal\dot{\Delta}\, w_1^\intercal] = \mathrm{Tr}[\dot{\Delta}\dot{\Delta}^\intercal] \geq 0$, where the dot-product denotes $\boldsymbol{a} \cdot \boldsymbol{b} \equiv \sum_i \mathrm{Tr}[a_i b_i^\intercal]$. We refer to this operation as Hessian$_+$ or $\boldsymbol{H}_+$.

Symmetry and conservation law Deep linear networks exhibit inherent symmetries that their input-output map $\bar{w}$ is invariant under the transformations that multiply an arbitrary matrix m to layer i and its inverse to the next layer $\begin{bmatrix}w_1 \\ w_2\end{bmatrix} \to \begin{bmatrix}mw_1 \\ w_2 m^{-1}\end{bmatrix}$, or under the equivalent continuous-time transformations: $\dot{\boldsymbol{w}}_{\text{null}} \equiv \begin{bmatrix}mw_1 \\ -w_2 m\end{bmatrix}$. Due to the invariance of the input-output map ($\dot{\Delta} = \dot{\bar{w}} = w_2 m_1 w_1 - w_2 m_1 w_1 = 0$) under these transformations, $\dot{\boldsymbol{w}}_{\text{null}}$ are orthogonal to gradient $\boldsymbol{g} \cdot \dot{\boldsymbol{w}}_{\text{null}} = \mathrm{Tr}[\Delta\dot{\Delta}^\intercal] = 0$, and also form the *null-space* of Hessian, since

$$\dot{\boldsymbol{w}}_{\text{null}} \cdot \boldsymbol{H}_+\dot{\boldsymbol{w}}_{\text{null}} = \mathrm{Tr}[\dot{\Delta}\dot{\Delta}^\intercal] = 0. \tag{4}$$

These continuous symmetries give rise to conservation laws (Noether's theorem):

$$d/dt\,(w_i w_i^\intercal - w_{i+1}^\intercal w_{i+1}) = 0 \tag{5}$$

which applies to all update dynamics $\dot{\boldsymbol{w}}$ that are orthogonal to the null-space: *i.e.* $\dot{\boldsymbol{w}} \cdot \dot{\boldsymbol{w}}_{\text{null}} = \sum_i \mathrm{Tr}[(w_i\dot{w}_i^\intercal - \dot{w}_{i+1}^\intercal w_{i+1})m_i] = 0$, including SGD and NGD updates.

2.1 Shallow network case ($d = 1$)

In case of shallow networks $\bar{w} = w_1$, the loss eq (1) reduces to the least-squares regression problem, whose SGD dynamics is linear (learning rate η)

$$\dot{\bar{w}} = -\eta g = -\eta(\bar{w} - w_*)\Sigma_x, \tag{6}$$

which exhibits mixture of exponentially converging dynamics. The time-constants of convergence are critically affected by the condition number of loss curvature, which in this case is the input correlations. Applying inverse Hessian neutralizes this effect ($\eta = 1/\tau$)

$$\dot{\bar{w}} = -\eta H^{-1} g = -(\bar{w} - w_*)/\tau, \tag{7}$$

yielding perfectly-conditioned convergence dynamics regardless of curvature. Therefore, for the linear learning dynamics of shallow networks, curvature correction merely normalizes the convergence time-constants, which is also achievable by simple whitening of input correlations.

In contrast, learning dynamics of deep networks exhibits complex nonlinearities due to the multiplicative weight coupling in the input-output map $\bar{w} = w_d \cdots w_1$ (Saxe et al., 2013), for which the effect of curvature correction remains obscure/unsolved. We investigate this problem by analyzing exact solutions to the learning dynamics of loss eq (1) under various curvature correction schemes. We assume pre-whitened input distribution to isolate the nonlinear phenomenon from the linear time-constant normalization effect mentioned above.

3 Learning dynamics of network weights

Deep networks learns to appropriately modify its overall function (*i.e.* the input-output map $\bar{w}$) by dynamically tuning their parameters (*i.e.* network weights) over the course of training. In this section, we analyze the learning dynamics of the network weights under SGD and the curvature corrected update rules, and in the next section, analyze its effect on the dynamics of input-output map.

Steepest gradient descent (SGD) SGD update dynamics of eq (1) for deep networks is given by ($d = 2$ example)

$$\dot{\boldsymbol{w}} + \eta \boldsymbol{g} = \begin{bmatrix} \dot{w}_1 + \eta\, w_2^\intercal\, \Delta \\ \dot{w}_2 + \eta\, \Delta\, w_1^\intercal \end{bmatrix} = \boldsymbol{0}, \tag{8}$$

where the update dynamics of layer i is driven by the displacement Δ and multiplied by weights of all other layers. For further analysis, eq (8) can be broken down via singular vector decomposition into simpler, decoupled dynamics of length d chains, with each chain representing one singular mode. Under the simplifying condition that the singular vectors of adjacent layers are *well aligned*[1], the learning dynamics of each singular mode chain is fully described by[2]

$$\dot{\sigma}_i = -\eta\, \sigma_\Delta\, r_i, \qquad (\sigma_\Delta \equiv \bar{\sigma} - \sigma_*,\ \bar{\sigma} = \prod_{i=1}^{d} \sigma_i,\ r_i = \prod_{j \neq i} \sigma_j) \tag{9}$$

where σ_i, σ_*, $\bar{\sigma}$, σ_Δ are the singular values of w_i, w_*, $\bar{w}$, Δ, and $r_i \equiv \partial \sigma_\Delta / \partial \sigma_i$ denotes the coupling between displacement σ_Δ and σ_i (See S.I.). This dynamics is more simply described in terms of its speed and direction: The direction of singular mode dynamics is prescribed by the conservation law (5) to follow the hyperbolic paths

$$\sigma_i^2 - \sigma_j^2 = \text{constant}, \tag{10}$$

and the speed is $\|\dot{\sigma}\| = \eta |\sigma_\Delta| \|r\|$, where $\|r\| \equiv \sqrt{\sum_{i=1}^{d} r_i^2}$ is the overall coupling strength. Normalizing the speed by the displacement yields the *effective learning rate* [any better name?]:

$$\eta_{\text{eff}} \equiv \frac{\|\dot{\sigma}\|}{|\sigma_\Delta|} = \eta\, \|r\|,$$

which vanishes/explodes for small/large coupling strength.

[1] Given the singular value decompositions $w_i = L_i A_i R_i^\intercal$, and $w_* = L_* A_* R_*^\intercal$, where the L/R are the orthogonal matrices of left/right singular vectors and A are the rectangular diagonal matrices of singular values, it is assumed that $\forall i, R_{i+1} = L_i$ and $L_* = L_d, R_* = R_1$, such that the input-output map $\bar{w} = w_d \cdots w_1 = L_d \left(\prod_{i=1}^{d} A_i\right) R_1^\intercal$ shares the same singular vectors with w_*. See (Saxe et al., 2013).

[2] Eq (9) applies upto N singular modes, where N is the narrowest width of the network (*i.e.* bottleneck size), which sets the number non-zero singular values at the bottleneck layer.

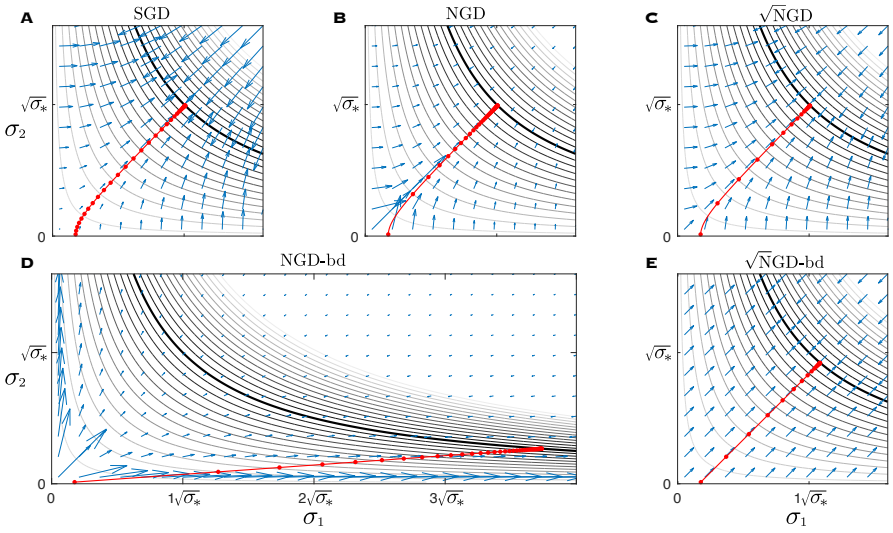

Figure 1: Learning dynamics of a singular mode in a depth $d = 2$ network (*i.e.* 1-hidden-layer). The stable manifold $\sigma_1\sigma_2 = \sigma_*$ is shown in black. The contour lines show the manifolds of constant displacement levels $\sigma_\Delta \equiv \sigma_1\sigma_2 - \sigma_*$, which are invariant to null-space transformations. The vector field visualizes the *displacement-normalized* update dynamics $[\dot\sigma_1, \dot\sigma_2]/|\sigma_\Delta|$, whose amplitude is the *effective learning rate*: $\eta_{\text{eff}} \equiv \|\dot\sigma\|/|\sigma_\Delta|$. (A,B,C) SGD, NGD, and $\sqrt{\text{N}}$GD share the same update directions, following hyperbolic paths (red line) that conserve $\sigma_1^2 - \sigma_2^2$, orthogonal to the contour lines. SGD exhibits vanishing rate problem for small weights $\{\sigma_i\}$, while NGD show the opposite problem. In contrast, $\sqrt{\text{N}}$GD exhibits constant effective learning rate. (D) NGD-bd exhibits radially diverging vector field that conserves σ_1/σ_2. (E) $\sqrt{\text{N}}$GD-bd exhibits vector field of constant direction and amplitude that conserves $|\sigma_1| - |\sigma_2|$.

Natural gradient descent (NGD) NGD is given by the Moore-Penrose (MP) pseudo-inverse[3] that finds the minimum-norm update ($\min \|\dot{\boldsymbol{w}}\|^2 \equiv \dot{\boldsymbol{w}} \cdot \dot{\boldsymbol{w}}$) subject to the constraint ($d = 2$ example)

$$\boldsymbol{H}_+\dot{\boldsymbol{w}} + \eta \boldsymbol{g} = \begin{bmatrix} w_2^\intercal(\dot\Delta + \eta\Delta) \\ (\dot\Delta + \eta\Delta)w_1^\intercal \end{bmatrix} = \boldsymbol{0}, \tag{11}$$

which can be solved using Lagrange multipliers

$$\begin{bmatrix} \dot w_1 + \eta\, w_2^\intercal\, \Lambda \\ \dot w_2 + \eta\, \Lambda\, w_1^\intercal \end{bmatrix} = \boldsymbol{0}, \tag{12}$$

where Λ satisfies the generalized Sylvester equation, $w_2^\intercal S(\Lambda) = S(\Lambda) w_1^\intercal = 0$ with $S(\Lambda) \equiv (w_2 w_2^\intercal)\Lambda + \Lambda(w_1^\intercal w_1) - \Delta$ (See S.I.). Remarkably, the only change from SGD update (8) is replacing Δ with Λ as the main drive of the update dynamics. Consequently, NGD update (12) shares many features with SGD, such as orthogonality to the null-space and the conservation law (5).

The singular mode analysis of eq (12) yields (See S.I.):

$$\dot\sigma_i = -\eta\, \sigma_\Delta\, \frac{r_i}{\|r\|^2}, \qquad (\|r\|^2 = \sum_{i=1}^{d} r_i^2) \tag{13}$$

which simply divides/normalizes the speed of singular mode dynamics eq (9) by $\|r\|^2$ while preserving the direction (Fig 1B). Therefore, NGD update follows the same hyperbolic path of SGD update, but with modified effective learning rate

$$\eta_{\text{eff}} = \frac{\eta}{\|r\|},$$

which explodes/vanishes for small/large coupling strength, reciprocal to SGD's problem.

[3] Due to the null-space of $\boldsymbol{H}_+$, the constraint eq (11) admits infinitely many solutions with arbitrary null-space components, since $\boldsymbol{H}_+(\dot{\boldsymbol{w}} + \dot{\boldsymbol{w}}_{\text{null}}) + \eta\boldsymbol{g} = \boldsymbol{H}_+\dot{\boldsymbol{w}} + \eta\boldsymbol{g} = \boldsymbol{0}$. MP-inverse yields the unique solution orthogonal to the null-space, equivalent to the limit of regularized inverse $\dot{\boldsymbol{w}} = -\eta \lim_{\epsilon\to 0}(\epsilon + 1)(\epsilon\boldsymbol{I} + \boldsymbol{H}_+)^{-1}\boldsymbol{g}$. The block-diagonal NGD eq (17)

Fractional Natural Gradient Descent ($\sqrt[q]{\text{N}}$GD) The above result can be generalized to a spectrum of learning rules that apply partial curvature corrections, described by $\sqrt[q]{\boldsymbol{H}_+}\dot{\boldsymbol{w}} + \eta \boldsymbol{g} = \boldsymbol{0}$, where $\sqrt[q]{\boldsymbol{H}_+}$ is a fractional power of Hessian ($q \geq 1$). This *fractional* NGD interpolates between NGD ($q \to 1$) and SGD ($q \to \infty$), with singular mode dynamics

$$\dot{\sigma}_i = -\eta \sigma_\Delta \frac{r_i}{\|r\|^{2/q}}, \tag{14}$$

which normalizes the update speed by $\|r\|^{2/q}$, while preserving the hyperbolic path shape. Note that at $q = 2$, termed $\sqrt{\text{N}}$GD, the effective learning rate becomes constant

$$\eta_{\text{eff}} = \eta, \tag{15}$$

thus neutralizing the vanishing/exploding problems of SGD and NGD (See Fig 1C).

Comparison to Regularized NGD Another interpolation can be obtained from solving $(\boldsymbol{H}_+\dot{\boldsymbol{w}} + \eta \boldsymbol{g}) + \epsilon(\dot{\boldsymbol{w}} + \eta \boldsymbol{g}) = \boldsymbol{0}$, which yields the regularized inverse $\dot{\boldsymbol{w}} = -\eta(\epsilon + 1)(\epsilon \boldsymbol{I} + \boldsymbol{H}_+)^{-1}\boldsymbol{g}$. The corresponding singular mode dynamics is

$$\dot{\sigma}_i = -\eta\, \sigma_\Delta \frac{r_i}{\|r\|} \frac{a\|r\| + 1}{a + \|r\|}, \qquad (a \equiv \epsilon/\|r\|) \tag{16}$$

where the ratio $a \equiv \epsilon/\|r\|$ describes the effective degree of interpolation between NGD ($a \to 0$) and SGD ($a \to \infty$). Note that a needs to be large enough in order to provide appropriate regularization for numerical stability, but it also cannot be too large to nullify the effect of curvature correction. Unlike q in $\sqrt[q]{\text{N}}$GD, however, this ratio a is not an explicit parameter, but an indirectly determined variable that constantly changes during learning and across singular modes. Remarkably, eq 16 reduces to $\sqrt{\text{N}}$GD at the mid-point ($a = 1$). In this sense, $\sqrt{\text{N}}$GD can be considered as an ideally-regularized NGD with hypothetical adaptive tuning $\epsilon = \|r\|$.

Block-diagonal Approximation of NGD (NGD-bd) In practical applications, numerically estimating and inverting Hessian of deep networks becomes prohibitively expensive. Instead, most second-order methods approximate NGD by applying *layer-restricted*, or *block-diagonal* curvature corrections (Martens & Grosse, 2015; Ba et al., 2016; Grosse & Martens, 2016; Martens et al., 2018; Bernacchia et al., 2018), in which the weight update of layer i only uses the Hessian term of the layer, discarding the off-diagonal curvature interactions with other layers: ($d = 2$ example)

$$\begin{bmatrix} H_1\dot{w}_1 + \eta_1 g_1 \\ H_2\dot{w}_2 + \eta_2 g_2 \end{bmatrix} = \begin{bmatrix} w_2^\intercal(w_2\dot{w}_1 + \eta_1\Delta) \\ (\dot{w}_2 w_1 + \eta_2\Delta)w_1^\intercal \end{bmatrix} = \boldsymbol{0}, \tag{17}$$

where H_i's denote the Hessian of layer i. This block-diagonal approximation (NGD-bd) introduces significant null-space component to the update dynamics, yet still satisfying the NGD constraint eq (11), given that $\sum_{i=1}^d \eta_i = \eta$. The singular mode dynamics of NGD-bd is (with $\eta_i = \eta/d$)

$$\dot{\sigma}_i = -\frac{\eta\, \sigma_\Delta}{d} \frac{r_i}{r_i^2} = -\frac{\eta\, \sigma_\Delta}{d} \frac{1}{r_i}, \tag{18}$$

where the layer-restricted factor r_i^2 substitutes NGD's full curvature correction factor $\|r\|^2$ in (13). Due to the non-zero null-space component, this dynamics deviates from the hyperbolic paths of NGD/SGD update and instead follows radially diverging paths that conserve the ratio σ_i/σ_j, (Fig 1D). As a result, NGD-bd finds less efficient solutions that require larger modification to converge (Fig 1D, red line).

Block-diagonal $\sqrt{\text{N}}$GD ($\sqrt{\text{N}}$GD-bd) More generally, block-diagonalization of $\sqrt[q]{\text{N}}$GD yields

$$\dot{\sigma}_i = -\frac{\eta\, \sigma_\Delta\, r_i}{(d\, r_i^2)^{1/q}}, \tag{19}$$

which conserves $\sigma_i^{2(1-1/q)} - \sigma_j^{2(1-1/q)}$ as constants of motion for $q > 1$. The effective learning rate is $\eta_{\text{eff}} = \eta\, \|r\|^{1-2/q}$. Note that for $q = 2$, called $\sqrt{\text{N}}$GD-bd, the singular mode dynamics follows non-diverging, straight parallel paths that conserve $|\sigma_i| - |\sigma_j|$, with constant $\eta_{\text{eff}} = \eta$, hence neutralizing the vanishing/exploding rate problems (See Fig 1E), analogous to $\sqrt{\text{N}}$GD.

4 LEARNING DYNAMICS OF INPUT-OUTPUT MAP

The previous section analyzed the update dynamics of weight parameters in deep networks. In this section, we investigate how the overall function of network, *i.e.* the input-output map $\bar{w}$, evolves during training. The map dynamics of NGD update is derived from eq (13)

$$\dot{\bar{\sigma}} = \sum_{i=1}^{d} r_i \dot{\sigma}_i = -\eta \, (\bar{\sigma} - \sigma_*) \, \frac{\sum_{i=1}^{d} r_i^2}{\|r\|^2} = -\eta(\bar{\sigma} - \sigma_*), \tag{20}$$

which is identical to the linear dynamics of shallow network learning eq (7). Moreover, since the map $\bar{w}$ is invariant to the null-space component of update, this result holds for NGD-bd and for any other generalized inverse solutions of the constraint eq (11) (Bernacchia et al., 2018).

More generally, the map dynamics of $\sqrt[q]{\text{N}}$GD is $\dot{\bar{\sigma}} = -\eta(\bar{\sigma} - \sigma_*)\|r\|^{2(1-1/q)}$, which, for the simplifying case of an identical singular value shared across all layers ($\forall i$, $\sigma_i = \bar{\sigma}^{1/d}$), reduces to

$$\dot{\bar{\sigma}} = -\bar{\eta} \, (\bar{\sigma} - \sigma_*) \, \bar{\sigma}^p \qquad \left(p \equiv \frac{2(d-1)(q-1)}{dq} \right) \tag{21}$$

where $\bar{\eta} \equiv \eta \, d^{1-1/q}$ is the *depth-calibrated* learning rate, and p represents the combined effect of depth and curvature correction that determines the *stiffness*, or numerical stability, of map dynamics.

Figure 2 shows the following notable closed-form solutions, as well as the $p = 2$ case:

$$\begin{aligned}
\bar{\sigma}_{(t)} &= \sigma_*(1 - e^{-\bar{\eta}t}) && (p = 0) \\
\bar{\sigma}_{(t)} &= \sigma_* \tanh^2(\bar{\eta}\sqrt{\sigma_*}t/2) && (p = 0.5) \\
\bar{\sigma}_{(t)} &= \frac{\sigma_*}{1 + (\sigma_*/\bar{\sigma}_{(0)} - 1)e^{-\bar{\eta}\sigma_* t}} && (p = 1)
\end{aligned}$$

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

NGD update ($q = 1$) The $p = 0$ case corresponds to shallow network learning ($d = 1$), as well as NGD update of arbitrarily deep networks, where the effect of depth is perfectly canceled out by curvature correction, such that map dynamics exhibits simple exponential convergence with constant time-scale $\bar{\eta}^{-1}$ across all singular modes. Consequently, the loss dynamics also exhibits exponentially decaying profiles: $L_{(t)} = L_{(0)}e^{-2\bar{\eta}t}$. Note that these loss profiles have finite slope $\dot{L}_{(t)} = -2\bar{\eta}L_{(t)}$ even as the gradient vanishes at $\|\boldsymbol{w}\| \to 0$, which is sustained by exploding the update norm $\|\dot{\boldsymbol{w}}\| \to \infty$.

SGD update ($q \to \infty$) For SGD update, $p = 1$ corresponds to training 1-hidden-layer networks, which exhibits sigmoidal learning curves. The learning time of singular modes scales with $(\bar{\eta}\sigma_*)^{-1}$, such that stronger modes (*i.e.* large σ_*) learn faster than weaker modes. The learning time also diverges as $\mathcal{O}(-\log \bar{\sigma}_{(0)})$ for small initial value $\bar{\sigma}_{(0)} \to 0$, due to vanishing gradient (Saxe et al., 2013). For deeper networks, the separation of time-scales and the slow rise of sigmoidal curves near zero intensifies as p increases with network depth, which approaches $p \to 2$ in infinite depth limit. The learning time scales with mode strength as $(\bar{\eta}\sigma_*^p)^{-1}$, and diverges as $\mathcal{O}(\bar{\sigma}_{(0)}^{1-p})$ for small initial values. This causes a sequence of learning from strong to weak singular modes in well-separated manner, which results in multiple stage-like transitions (plateaus) of loss-profile over the course of training.

$\sqrt{\text{N}}$GD update ($q = 2$) The $p = 0.5$ case corresponds to training 1-hidden-layer networks under $\sqrt{\text{N}}$GD update, which facilitates map dynamics to exhibit much smoother convergence than the sigmoidal profiles of SGD update. Even for deeper networks, the map dynamics under $\sqrt{\text{N}}$GD update is strictly less stiff than SGD training of 1-hidden-layer networks, since p approaches 1 only in infinite depth limit. Moreover, all $p < 1$ cases exhibit polynomial learning curves near zero, $\bar{\sigma}(t) \approx ((1-p)\bar{\eta}\,\sigma_* t)^{1/(1-p)}$, which therefore escape from zero initial condition $\bar{\sigma}_{(0)} \to 0$ in finite time. This is due to $\sqrt{\text{N}}$GD neutralizing the vanishing/exploding update problem and maintaining a constant effective learning rate η_{eff}.

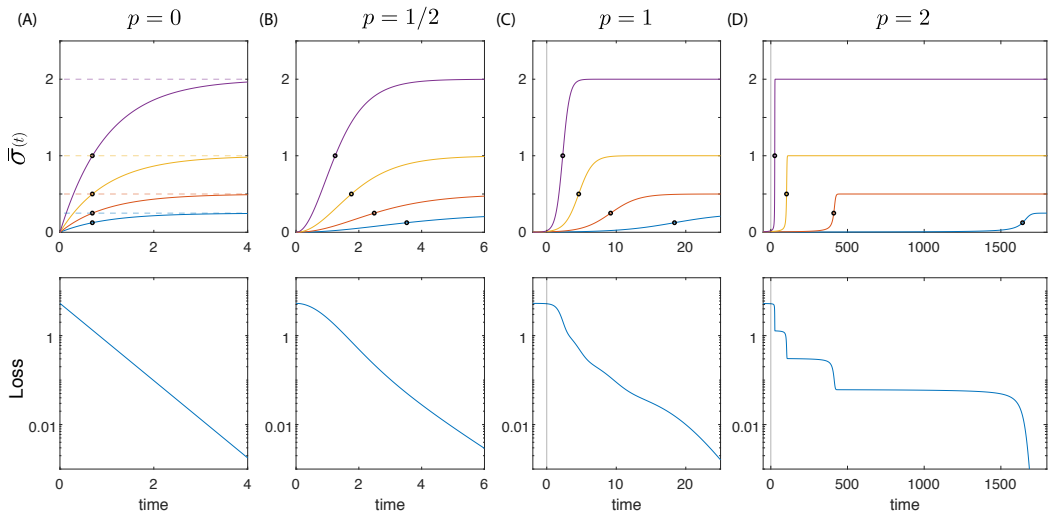

Figure 2: Top: Learning curves of map singular modes $\bar{\sigma}_{(t)}$ for various stiffness numbers p. The input-output correlations for each mode σ_* is shown by dashed lines. Half-max points (black circles) are shown to visualize the time-scale of learning, which scales as σ_*^{-p}. Bottom: Corresponding loss profiles. Initial conditions: $\bar{\sigma}_{(0)} = 0$ for $p < 1$, and $\bar{\sigma}_{(0)} = \sigma_*/100$ for $p \geq 1$. $\bar{\eta} = 1$.

Effective Depth Note that d and q contribute to stiffness in a symmetric manner. This relationship can be intuitively understood by representing stiffness in terms of the corresponding network depth under SGD update, called the *effective depth*:

$$d_{\text{eff}} = \frac{dq}{d + q - 1}, \tag{22}$$

which approaches the actual depth d in the SGD limit ($q \to \infty$), and similarly, approaches q in the limit of infinite depth ($d \to \infty$). Therefore, the input-output map learning dynamics of finite depth d network under $\sqrt[q]{\text{N}}$GD update exhibits the same level of stiffness as the SGD update of depth d_{eff} network, which is strictly less than q.

5 Effects on Generalization

Recent works have shown that SGD update has implicit regularization, which allows learning on training dataset to generalize well to testing set. However, it has not been shown whether such regularization property generalizes to update rules with curvature correction.

A crucial insight from recent works on generalization dynamics suggest that deep networks with small weight initializations can avoid overfitting via early stopping because they first learn the relevant signal dimensions of the dataset before the irrelevant noise dimension begins to fit (Advani & Saxe, 2017; Lampinen & Ganguli, 2018). Here, we test this result in the student-teacher task: The training and test dataset is generated by a teacher network $y^\mu = w_* x^\mu + z^\mu$, where $x^\mu \in \mathbb{R}^N$ is the input data, $y^\mu \in \mathbb{R}^N$ is the output, $w_* x^\mu$ is the signal and $z^\mu \in \mathbb{R}^N$ is the noise. The teacher's mapping $w_* \in \mathbb{R}^{N \times N}$ has a low-rank structure (rank 3), and the student network is a depth $d = 4$ network of constant width N, whose weight matrices are initialized to be orthogonal matrices with scaling factor of $1/40$. The number of training dataset $\{x^\mu, y^\mu\}_{\mu=1}^P$ is set to be equal to the effective number free parameters $P = N$, which makes learning most susceptible to overfitting.

This experiment used full-batch training. Hessian$_+$ is numerically estimated from the training data and inverted (sqrt-inverted) via SVD for NGD ($\sqrt{\text{N}}$GD) updates. Because of the discrete-time update, the weight updates $\delta \boldsymbol{w}$ under NGD and SGD rules are clipped to avoid the exploding rate problems. $\sqrt{\text{N}}$GD does not require such clipping.

Figure 3 shows the result of training: As previously described, under SGD update, the network learns the three signal modes first, well separated from the onset of overfitting of the noise modes

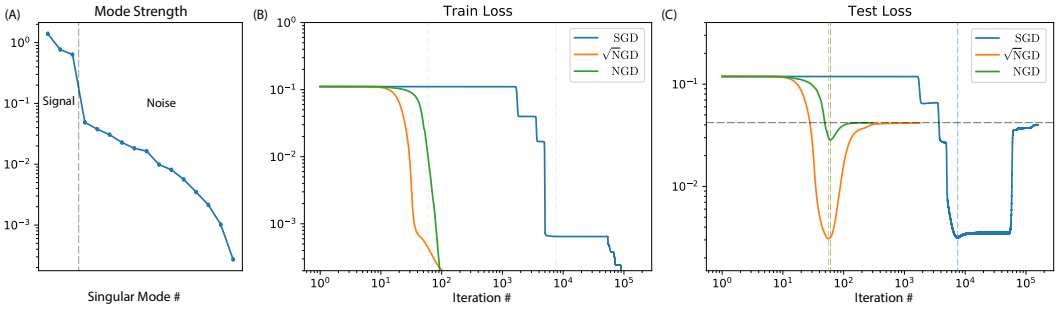

Figure 3: Curvature correction effects on generalization : (A) Singular mode strength of input-output correlation of a training dataset. Dataset is generated from a rank-3 teacher network with added noise (SNR = 10). (B, C) Training and testing loss profiles of a 3-hidden-layer student network. Vertical dashed-lines show the optimal early stopping time for each update rule.

begins (vertical dashed lines) for high SNR cases, which allows effective early stopping scheme. However, the long plateaus which allows the separates the time scales of signal and noise modes, also prolongs the overall duration of training.

NGD and $\sqrt{\text{N}}$GD exhibit much faster learning dynamics. NGD update, however, makes all modes to learn simultaneously, including the noise modes, which cannot be separated from the signal mode learning. Therefore, NGD update leads to high generalization error even at optimal early stopping time. Note that NGD's loss profile deviates from exponential decay due to the clipping.

In contrast, $\sqrt{\text{N}}$GD allows separation between the signal and the noise modes, since it scales the learning-time of each singular mode according to with mode strength. Consequently, $\sqrt{\text{N}}$GD update can achieve comparable test loss as SGD update, but also with fast early-stopping time comparable to NGD update. Note that all three update rules achieve the same test loss after overfitting is complete. This IS BECAUSE OF they all take the same learning path.

6 Discussion

A critical result of our analysis is that curvature correction maintains critical properties of SGD weight updates, *i.e.* orthogonality to null-space of Hessian, which generalizes the conservation law eq (5) to curvature-corrected update rules. As a result, along each singular mode chain, curvature correction only affects the speed/temporal profile of learning, while preserving the direction/path unchanged. This result may seem surprising, because in shallow network, because curvature correction is usually associated with changing of update direction, by changing the metric.

Optimal choice of curvature correction Here we discussed the idea of stiffness in learning dynamics, which adversely affects the numerical integration of the dynamics equation. Without curvature correction, steepest gradient has problem navigating through the convoluted space of loss landscape, with vanishing gradient being the prominent manifestation of the problem. In terms of map dynamics, NGD proposes the best solution for correcting the ill-conditioned curvature problem, which, in case of deep linear network, indeed completely resolves the nonlinearities of learning dynamics. In term of parameter dynamics, however, NGD has several problems. Accurate estimation of Hessian is difficult in moderate batch-size, and such estimation error would be further amplified when inverting the highly ill-conditioned Hessian. Therefore, most methods require large batch size and heavy/sophisticated application of gradient clipping and Hessian damping in order to reduce the noise level, which however, would, inevitably reduce the effectiveness of curvature correction. Moreover, as our analysis shows, the layer-wise curvature correction via block-diagonal Hessian approximation, which is unavoidable for most practical usage, produces update direction that is significantly different from true natural gradient direction and divergent in nature, although it would require further investigation to understand how this affects the network properties in nonlinear settings.

Furthermore, the separation of time scales in SGD has been suggested to have critical implications for learning: It has been proposed to have relevance for similarity between deep learning and human perceptual learning (Saxe et al., 2019), as well as for optimal early stopping time to maximize generalization of deep learning, which allows the meaningful signal features of the data to be learned before the onset of overfitting the small, noise features (assumed small) begins (Advani & Saxe, 2017; Lampinen & Ganguli, 2018). Therefore, NGD's advantage of and it maybe prone to overfitting of noisy data, because it does not separate time scales of learning. Moreover, the block-diagonal approximations of NGD shows very different update dynamics and tend to amplify the weight differences across layers during training.

We propose $\sqrt{\text{N}}$GD as an alternative optimization method that merges the advantages of both SGD and NGD. It allows training from zero-initial weights within finite time, yet allows separation of time scales that may benefit generalization properties.

REFERENCES

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

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

Jimmy Ba, Roger Grosse, and James Martens. Distributed second-order optimization using kronecker-factored approximations. 2016.

Alberto Bernacchia, Mate Lengyel, and Guillaume Hennequin. 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.

Aleksandar Botev, Hippolyt Ritter, and David Barber. 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.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pp. 2933–2941, 2014.

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

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

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

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

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

James Martens, Jimmy Ba, and Matt Johnson. Kronecker-factored curvature approximations for recurrent neural networks. 2018.

Nicolas L Roux, Pierre-Antoine Manzagol, and Yoshua Bengio. Topmoumoute online natural gradient algorithm. In *Advances in neural information processing systems*, pp. 849–856, 2008.

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

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1820226116. URL

Exact analysis of curvature corrected learning dynamics in deep linear networks - Supplementary Materials

Anonymous Author(s)
Affiliation
Address
email

1 Singular mode analysis of SGD update: eq (6) in Section 3.1

The SGD update :

$$\dot{\boldsymbol{w}} + \eta \boldsymbol{g} = \begin{bmatrix} \dot{w}_1 + \eta w_2^\intercal \Delta \\ \dot{w}_2 + \eta \Delta w_1^\intercal \end{bmatrix} = \mathbf{0}, \tag{1}$$

assumption: aligned singular vectors:

$w_1 = \sum_m \vec{u}_1^m \sigma_1^m \vec{u}_0^{m\intercal}$, $w_2 = \sum_m \vec{u}_2^m \sigma_2^m \vec{u}_1^{m\intercal}$, $w^* = \sum_m \vec{u}_2^m \sigma_*^m \vec{u}_0^{m\intercal}$.

where $\vec{u}_k^m$ are orthonormal vectors (singular vectors) that satisfy $\vec{u}_k^{m\intercal} \vec{u}_k^m = \delta_{nm}$, and σ_1^m, σ_2^m, σ_*^m are the singular values.

The displacement ($\Delta \equiv w_2 w_1 - w^*$) can be expressed as $\Delta = \sum_m \vec{u}_2^m \sigma_\Delta^m \vec{u}_0^{m\intercal}$, where $\sigma_\Delta^m \equiv \sigma_2^m \sigma_1^m - \sigma_*^m$.

The first layer portion of eq (1) can be expressed as

$$\sum_m \vec{u}_1^m (\dot{\sigma}_1^m + \eta \sigma_2^m \sigma_\Delta^m) \vec{u}_0^{m\intercal} + \dot{\vec{u}}_1^m \sigma_1^m \vec{u}_0^{m\intercal} + \vec{u}_1^m \sigma_1^m \dot{\vec{u}}_0^{m\intercal} = 0 \tag{2}$$

Multiplying eq (5) by the orthogonal vectors yields:

$$\begin{aligned} &\vec{u}_1^{n\intercal} \left(\sum_m \vec{u}_1^m (\dot{\sigma}_1^m + \eta \sigma_2^m \sigma_\Delta^m) \vec{u}_0^{m\intercal} + \dot{\vec{u}}_1^m \sigma_1^m \vec{u}_0^{m\intercal} + \vec{u}_1^m \sigma_1^m \dot{\vec{u}}_0^{m\intercal} \right) \vec{u}_0^n \\ &= \dot{\sigma}_1^n + \eta \sigma_2^n \sigma_\Delta^n \\ &= \dot{\sigma}_1^n + \eta \frac{\bar{\sigma}^n \sigma_\Delta^n}{\sigma_1^n} \\ &= 0 \end{aligned}$$

where we used $\vec{u}_k^{m\intercal} \vec{u}_k^m = \delta_{nm}$ and $\vec{u}_k^{m\intercal} \dot{\vec{u}}_k^m = 0$, and $\bar{\sigma}^n \equiv \Pi_i \sigma_i^n$. is the singular values of the input-output map $\bar{w}$. This result generalizes to networks of any depth D: For layer i,

$$\dot{\sigma}_i^m = -\eta \frac{\bar{\sigma}^m \sigma_\Delta^m}{\sigma_i^m} = -\eta \frac{\bar{\sigma}^m}{\sigma_i^m} (\bar{\sigma}^m - \sigma_*^m).$$

In the main text, we drop the singular mode index m for notational simplicity.

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

2 Moore-Penrose inverse solution: eq (15,16,17) in Section 3.2

In section 4, we find the Moore-Penrose inverse solution of the natural gradient condition:

$$\boldsymbol{H}\dot{\boldsymbol{w}} + \eta \boldsymbol{g} = \begin{bmatrix} w_2^\intercal(\dot{\Delta} + \eta\Delta) \\ (\dot{\Delta} + \eta\Delta)w_1^\intercal \end{bmatrix} = \boldsymbol{0}, \tag{3}$$

which minimizes the Frobenius norm of $\dot{\boldsymbol{w}}$ while satisfying the constraint. This constrained optimization problem is described by the following Lagrangian:

$$\mathcal{L}(\dot{w}_1, \dot{w}_2, \Lambda_2, \Lambda_2) = (\dot{w}_1 \cdot \dot{w}_1 + \dot{w}_2 \cdot \dot{w}_2)/2 + \Lambda_1 \cdot w_2^\intercal(\dot{\Delta} + \eta\Delta) + \Lambda_2 \cdot (\dot{\Delta} + \eta\Delta)w_1^\intercal,$$

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

$$\partial\mathcal{L}/\partial\dot{w}_1 = \dot{w}_1 + w_2^\intercal w_2\Lambda_1 + w_2^\intercal\Lambda_2 w_1 = 0 \tag{4}$$

$$\partial\mathcal{L}/\partial\dot{w}_2 = \dot{w}_2 + w_2\Lambda_1 w_1^\intercal + \Lambda_2 w_1 w_1^\intercal = 0 \tag{5}$$

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

$$\dot{w}_1 + \eta w_2^\intercal\Lambda = 0 \tag{6}$$

$$\dot{w}_2 + \eta\Lambda w_1^\intercal = 0 \tag{7}$$

which can be plugged into the optimality condition on Λ_i

$$\partial\mathcal{L}/\partial\Lambda_1 = w_2^\intercal(\dot{\Delta} + \eta\Delta) = 0 \tag{8}$$

$$\partial\mathcal{L}/\partial\Lambda_2 = (\dot{\Delta} + \eta\Delta)w_1^\intercal = 0 \tag{9}$$

to produce a linear equation for Λ_i:

$$w_2^\intercal S(\Lambda) = S(\Lambda)w_1^\intercal = 0 \tag{10}$$

$$\text{where} \quad S(\Lambda) = (w_2 w_2^\intercal)\Lambda + \Lambda(w_1^\intercal w_1) - \Delta. \tag{11}$$

Note that eq (10) reduces to Sylvester equation, $S(\Lambda) = 0$, if w_2, w_1 are full rank. Therefore, eq (10) can be understood as the pseudo-inverse version of Sylvester equation for rank deficient problems.

The Moore-Penrose inverse $\dot{\boldsymbol{w}}_{\text{MP}}$ is the unique solution of natural gradient that is orthogonal to the null space. All other *generalized inverse* solutions to natural gradient condition, including block-diagonal natural gradient, differ from $\dot{\boldsymbol{w}}_{\text{MP}}$ only in their null space components, since

$$(\boldsymbol{H}\dot{\boldsymbol{w}}_{\text{MP}} + \eta\boldsymbol{g}) - (\boldsymbol{H}\dot{\boldsymbol{w}} + \eta\boldsymbol{g}) = \boldsymbol{H}(\dot{\boldsymbol{w}}_{\text{MP}} - \dot{\boldsymbol{w}}) = \boldsymbol{0} \tag{12}$$

3 Singular mode analysis eq (18) in Section 3.2

We follow the approach of [1] and consider well-aligned singular vector condition, which allows analyzing the dynamics of individual singular modes independently. In the following analysis, we introduce $\sigma_i, \bar{\sigma}, \sigma_\Delta, \sigma_\Lambda, \sigma_S$ which represent the singular values of $w_i, \bar{w}, \Delta, \Lambda, S(\Lambda)$ of one singular mode.

In this representation, eq (6) and (7) reduce to

$$\dot{\sigma}_i = -\sigma_\Lambda \frac{\bar{\sigma}}{\sigma_i} \tag{13}$$

whereas, eq (10) and (11) reduce to

$$\sigma_i\sigma_S = 0 \tag{14}$$

$$\sigma_S = \sum_{i=1}^{D} \left(\frac{\bar{\sigma}}{\sigma_i}\right)^2 \sigma_\Lambda - \eta\sigma_\Delta \tag{15}$$

Since we only consider trainable singular modes (*i.e.* $\forall i, \sigma_i \neq 0$), eq (14) implies $\sigma_S = 0$, which reduces eq (15) to

$$\sigma_\Lambda = \eta \frac{\sigma_\Delta}{\sum_{i=1}^{D} (\bar{\sigma}/\sigma_i)^2} \tag{16}$$

which plugs into (13) to produce the result in the main text

$$\dot{\sigma}_i = -\eta \frac{\sigma_\Delta}{\sum_{i=1}^{D} (\bar{\sigma}/\sigma_i)^2} \frac{\bar{\sigma}}{\sigma_i} \tag{17}$$

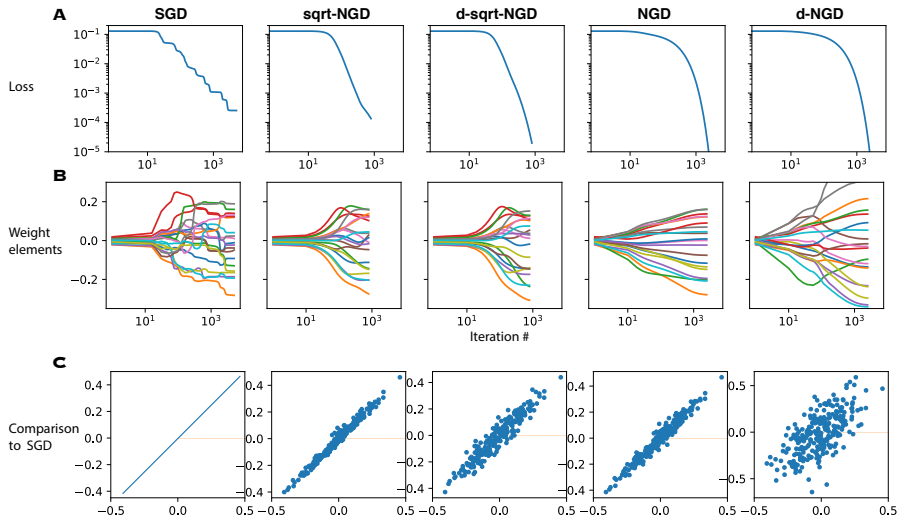

Figure 1: Training of a linear 4-layer ($D = 3$) *student network* that learns to recover the mappings defined by a random *teacher network*. Identical initial weights were used for all student networks and trained by various learning methods. (A) Loss profile: SGD training shows multiple plateaus in the loss profile due to separation of time scales. $\sqrt{\text{N}}$GD and d-$\sqrt{\text{N}}$GD training show smoother and faster convergence without multiple plateaus, yet exhibit a level of time-scale separation, where small noise modes are learned at the end. NGD and d-NGD show exponential convergence without separation of time scales. (B) Evolution of a randomly sampled set of weight elements during training: SGD training exhibits complex nonlinear dynamics in weights due to multiple plateaus. $\sqrt{\text{N}}$GD and d-$\sqrt{\text{N}}$GD shows smoother dynamics that are almost identical to each other, and exhibit some similarity to SGD dynamics. NGD trajectories exhibit smooth, exponential convergence. d-NGD, however, exhibits very different, diverging trajectories, even though its loss profile is identical to NGD profile. (C) Comparison of final weight element values at the end of training. x-axis: SGD final weight, y-axis: final weights from other learning methods.

4 Weight trajectory of during learing

In main text, we showed that SGD, NGD and $\sqrt{\text{N}}$GD all conserve the same constants of motion and thus follow the same path of learning per singular mode, and they differ only in their temporal profiles of learning. This result implies that all of them should converge to the same point on the solution manifold given the same initial condition, even though their learning trajectories may differ. Figure 1C here indeed confirms this prediction. As predicted, the final weights of NGD and $\sqrt{\text{N}}$GD are very close to those of SGD training. The small differences can be attributed to using finite update step-size, instead of continuous time version. d-$\sqrt{\text{N}}$GD final weights show more deviations but still show strong similarities to SGD result. d-NGD final weights show very little correlation.

5 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 [1], 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.

Standard SGD training, standard pytorch SGD optimizer was used. For d-NGD training, we used the block-diagonal, Kronecker-factored approximation of hessian, similar to KFAC algorithm, implemented in pytorch, which uses generalized Gauss-Newton matrix as the preconditioner instead of Fisher matrix. SVD (singular value decomposition) was used to invert the layer-wise hessian (with added damping). The same algorithm was also used for d-$\sqrt{\text{N}}$GD training, except that it inverted the

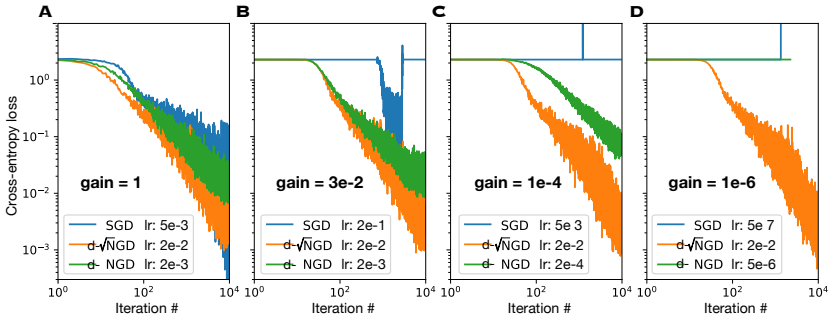

Figure 2: Reviving vanishing gradient: Training a 5-layer ReLU network on MNIST dataset. Weights are initialized to be orthonormal matrices with various gains that range between 1 and 10^{-6}. NGD-bd requires a small damping term for inverting Hessian $(\epsilon \boldsymbol{I} + \boldsymbol{H}_+)^{-1}$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

square-root of the hessian's singular values (without damping). For numerical stability, the amount of damping and learning rate had to change for different initial weight gains for d-NGD training. In contrast, d-$\sqrt{\text{N}}$GD training was unaffected by the initial weight gain.

References

[1] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.