SYMMETRY, CONSERVATION LAWS, AND LEARNING DYNAMICS IN NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting the dynamics of neural network parameters during training is one of the key challenges in building a theoretical foundation for deep learning. A central obstacle is that the motion of a network in high-dimensional parameter space undergoes discrete finite steps along complex stochastic gradients derived from real-world datasets. We circumvent this obstacle through a unifying theoretical framework based on intrinsic symmetries embedded in a network's architecture that are present for *any* dataset. We show that any such symmetry imposes stringent geometric constraints on gradients and Hessians, leading to an associated conservation law in the continuous-time limit of stochastic gradient descent (SGD), akin to Noether's theorem in physics. We further show that finite learning rates used in practice can actually break these symmetry induced conservation laws. We apply tools from finite difference methods to derive *modified gradient flow*, a differential equation that better approximates the numerical trajectory taken by SGD at finite learning rates. We combine modified gradient flow with our framework of symmetries to derive exact integral expressions for the dynamics of certain parameter combinations. We empirically validate our analytic predictions for learning dynamics on VGG-16 trained on Tiny ImageNet. Overall, by exploiting symmetry, our work demonstrates that we can analytically describe the learning dynamics of various parameter combinations at finite learning rates and batch sizes for state of the art architectures trained on *any* dataset.

1 INTRODUCTION

Just like the fundamental laws of classical and quantum mechanics taught us how to control and optimize the physical world for engineering purposes, a better understanding of the laws governing neural network learning dynamics can have a profound impact on the optimization of artificial neural networks. This raises a foundational question: *what, if anything, can we quantitatively predict about the training dynamics of large scale deep learning models driven by real-world datasets?* The central challenge to answering this question is managing the complexity of the stochastic gradients and finite step sizes that drive learning. In this work, we disentangle the complexity introduced by the dataset from geometric properties embedded in a network's architecture that do not depend on data. To do this, we use the lens of *symmetry* to show that if the training loss doesn't change under some transformation of the parameters, then the gradient and Hessian for those parameters have associated geometric constraints. We systematically apply this approach to modern neural networks to derive exact integral expressions for the dynamics of parameter combinations during training and verify our predictions empirically on large scale models and datasets. Our main contributions are:

1. We unify previously discovered geometric constraints on neural network gradients and Hessians by leveraging intrinsic translation, scale, and inversion symmetries (Sec. 3).
2. We show that these geometric constraints are data-independent, thereby constraining SGD noise to low-rank subspaces for *any* dataset (Sec.5).
3. We show that each of these symmetries has an associated *constant of motion*, which in the continuous time limit of stochastic gradient descent (SGD) is conserved (Sec.4).
4. We demonstrate under weight decay that these previously conserved quantities decay exponentially, thereby providing an account of the impact of weight decay on training (Sec.4).
5. We apply tools from finite difference methods to introduce a stochastic differential equation (SDE) with modified gradients to model SGD even with finite-learning rates. (Sec.5).

6. We use this SDE to demonstrate that finite learning rates break continuous-time conversation laws, and find exact solutions for the dynamics of quantities that were previously constant (Sec.6).
7. We validate our exact solutions empirically on VGG-16 with and without batch nornalization and trained on Tiny ImageNet (Sec.6).

2 Related work

Gradient geometry. A wide range of literature has discussed constraints on gradients originating from specific architectural building blocks of networks. For the first part of our work, we build on this literature and unify it through the framework of symmetry.

The earliest works understanding the importance of invariances in neural networks come from the loss landscape literature Baldi & Hornik (1989) and the characterization of critical points in the presence of explicit regularization Kunin et al. (2019). More recent works have studied implicit regularization originating from linear Arora et al. (2018b) and homogeneous Du et al. (2018) activations, finding that gradient geometry plays an important role in biasing the training dynamics. A different line of research studying the generalization capacity of networks have noticed similar gradient structures Liang et al. (2019). Beyond theoretical studies, geometric properties of the gradient and Hessian have been applied to optimize Neyshabur et al. (2015), interpret Bach et al. (2015), and prune Tanaka et al. (2020) neural networks.

Gradient properties introduced by batch Ioffe & Szegedy (2015), weight Salimans & Kingma (2016) and layer Ba et al. (2016) normalization have been intensely studied. Van Laarhoven (2017); Zhang et al. (2018) showed that normalization layers are scale invariant, but have an implicit role in controlling the effective learning rate. Cho & Lee (2017); Hoffer et al. (2018); Chiley et al. (2019); Li & Arora (2019); Wan et al. (2020) have leveraged the scale invariance of batch normalization to understand geometric properties of the learning dynamics.

Modeling SGD. To make experimentally testable predictions on learning dynamics, we introduce a continuous model for stochastic gradient descent (SGD). There exists a large body of works studying this subject using stochastic differential equations (SDEs) in the continuous-time limit Mandt et al. (2015; 2017); Li et al. (2017); Smith & Le (2017); Chaudhari & Soatto (2018); Jastrzębski et al. (2017); Zhu et al. (2018); An et al. (2018). Each of these works involves making specific assumptions on the loss or the noise in order to derive stationary distributions. More careful treatment of stochasticity led to fluctuation dissipation relationships at steady state without such assumptions Yaida (2018). In our analysis, we apply a more recent approach Li et al. (2017); Barrett & Dherin (2020), inspired by finite difference methods, that augments SDE model with higher-order terms to account for the effect of a finite step size and curvature in the learning trajectory.

3 Symmetries in the Loss shape Gradient and Hessian Geometries

While we initialize neural networks randomly, their gradients and Hessians at all points in training, no matter the loss or dataset, obey certain geometric constraints. Some of these constraints have been noticed previously as a from of implicit regularization, while others have been leveraged algorithmically in applications from network pruning to interpretability. Surprisingly, these geometric constraints can all be understood as consequences of numerous differentiable symmetries in the loss introduced by neural network architectures. A set of parameters observes a differentiable symmetry in the loss if the loss doesn't change under a certain differentiable transformation of these parameters. This invariance introduces associated geometric constraints on the gradient and Hessian.

Consider a function $\mathcal{R}(\theta)$. This function possesses a symmetry if it is invariant under the action ψ of a group G on the parameter vector θ : i.e. if $\theta \to \psi(\theta, \alpha)$ where $\alpha \in G$, then $\mathcal{R}(\theta) = \mathcal{R}(\psi)$ for all $\alpha \in G$. This symmetry implies $\frac{d\mathcal{R}}{d\alpha} = \langle \frac{\partial \mathcal{R}}{\partial \psi(\theta)} \frac{\partial \psi(\theta)}{\partial \alpha} \rangle = 0$. Evaluating this expression at identity element $\alpha = I$ of G, so that $\psi = \theta$ then yields the result that the gradient $\frac{\partial \mathcal{R}}{\partial \theta}$ is perpendicular to the vector field $\frac{\partial \psi(\theta)}{\partial \alpha}$ that generates the symmetry, for all θ. To make this analysis concrete, we consider three families of symmetries, translation, scale, and inversion, that commonly appear in modern network architectures.

3.1 TRANSLATION

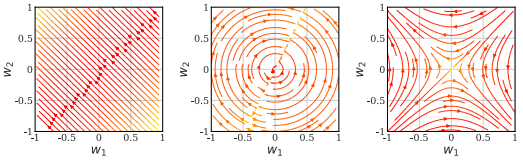
Translation symmetry is defined by the group $\mathbb{R}$ and action $\theta \mapsto \theta + \alpha\mathbb{1}$. The loss function $\mathcal{L}(\theta)$ thus possesses translation symmetry if $\mathcal{L}(\theta) = \mathcal{L}(\theta + \alpha\mathbb{1})$ for all $\alpha \in \mathbb{R}$. Such a symmetry in turn implies the loss gradient is orthogonal to the all-ones vector:

$$\langle \tfrac{\partial \mathcal{L}}{\partial \theta}, \mathbb{1} \rangle = 0. \tag{1}$$

Note the vector valued function $\mathcal{R}(\theta) = \frac{\partial \mathcal{L}}{\partial \theta}$ is also translation invariant. The implied orthogonality condition on the gradient of $\mathcal{R}$ then implies a constraint on the Hessian H

$$H\mathbb{1}_\theta = 0, \tag{2}$$

where $\mathbb{1}_\theta$ is the indicator vector for the subset of parameters with translation symmetry.



(a) Translation (b) Scale (c) Inversion

Figure 1: **Visualizing symmetry.** We visualize the vector fields associated with simple network components that have translation, scale, and inversion symmetry. In (a) we consider the vector field associated with a neuron $\sigma\left([w_1 \quad w_2]^\intercal x\right)$ where σ is the softmax function. In (b) we consider the vector field associated with a neuron $\mathrm{BN}\left([w_1 \quad w_2]\,[x_1 \quad x_2]^\intercal\right)$ where BN is the batch normalization function. In (c) we consider the vector field associated with a linear path $w_2 w_1 x$.

Softmax function. Any network using the softmax function gives rise to translation symmetry for the parameters immediately preceding the function. Let $z = Wx + b$ be the input to the softmax function such that $\sigma(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$. Notice that shifting any column of the weight matrix w_i or the bias vector b by a real constant has no effect on the output of the softmax as the shift factors from both the numerator and denominator canceling its effect. Thus, the loss function is invariant w.r.t. this translation, yielding gradient constraints $\langle \frac{\partial \mathcal{L}}{\partial w_i}, \mathbb{1} \rangle = \langle \frac{\partial \mathcal{L}}{\partial b}, \mathbb{1} \rangle = 0$.

3.2 SCALE

Scale symmetry is defined by the group $\mathrm{GL}_1(\mathbb{R})$ and action $\theta \mapsto \alpha\theta$. The loss function possesses scale symmetry if $\mathcal{L}(\theta) = \mathcal{L}(\alpha\theta)$ for all $\alpha \in \mathrm{GL}_1(\mathbb{R})$. This symmetry immediately implies the loss gradient is everywhere perpendicular to the parameter vector itself:

$$\langle \tfrac{\partial \mathcal{L}}{\partial \theta}, \theta \rangle = 0. \tag{3}$$

Note the vector function $\mathcal{R}(\theta) = \frac{\partial \mathcal{L}}{\partial \theta} \odot \theta$ also possesses scale symmetry. The implied orthogonality condition on the gradient of $\mathcal{R}$ then implies a constraint on the loss gradient and Hessian

$$H\theta + \tfrac{\partial \mathcal{L}}{\partial \theta} = 0. \tag{4}$$

Batch normalization. Batch normalization leads to scale invariance during training. Let $z = w^\intercal x + b$ be the input to a neuron with batch normalization such that $\mathrm{BN}(z) = \frac{z - \mathrm{E}[z]}{\sqrt{\mathrm{Var}(z)}}$ where $\mathrm{E}[z]$ is the sample mean and $\mathrm{Var}(z)$ is the sample variance given a batch of data. Notice that scaling w and b by a non-zero real constant has no effect on the output of the batch normalization as it factors from z, $\mathrm{E}[z]$, and $\mathrm{Var}(z)$ canceling its effect. Thus, these parameters observe scale symmetry in the loss and their gradients satisfy $\langle \frac{\partial \mathcal{L}}{\partial w}, w \rangle + \langle \frac{\partial \mathcal{L}}{\partial b}, b \rangle = 0$, as has been previously noted by Ioffe & Szegedy (2015); Van Laarhoven (2017).

3.3 INVERSION

Inversion symmetry is defined by the group $\mathrm{GL}_1(\mathbb{R})$ and action $(\theta_1, \theta_2) \mapsto (\alpha\theta_1, \alpha^{-1}\theta_2)$. Thus the loss function possesses scaling symmetry if $\mathcal{L}(\theta_1, \theta_2) = \mathcal{L}(\alpha\theta_1, \alpha^{-1}\theta_2)$ for all $\alpha \in \mathrm{GL}_1(\mathbb{R})$. The gradient of the loss with respect to the parameters satisfies the property,

$$\langle \tfrac{\partial \mathcal{L}}{\partial \theta_1}, \theta_1 \rangle - \langle \tfrac{\partial \mathcal{L}}{\partial \theta_2}, \theta_2 \rangle = 0. \tag{5}$$

Note the function $\mathcal{R}(\theta) = \frac{\partial \mathcal{L}}{\partial \theta} \odot (\theta_1 + \theta_2)$ also possess inversion symmetry. The implied orthogonality conditions on the gradients of $\mathcal{R}$ imply the Hessian constraint

$$H\theta_1 - H\theta_2 + \tfrac{\partial \mathcal{L}}{\partial \theta_1} - \tfrac{\partial \mathcal{L}}{\partial \theta_2} = 0. \tag{6}$$

Homogeneous activation. For networks with continuous, homogeneous activation functions $\phi(x) = \phi'(x)x$ (e.g. ReLU, Leaky ReLU, linear), this symmetry emerges at every hidden neuron by considering all incoming and outgoing parameters to the neuron. For example, consider a hidden neuron with ReLU activation and let $w_2\text{ReLU}(w_1^\intercal x + b)$ be the computational path through this neuron. Scaling w_1 and b by a real constant and w_2 by its inverse has no effect on the computational path as the constants can be passed through the ReLU activation canceling their effects. Thus, these parameters observe inversion symmetry and their gradients satisfy $\left\langle \frac{\partial \mathcal{L}}{\partial w_1}, w_1 \right\rangle + \left\langle \frac{\partial \mathcal{L}}{\partial b}, b \right\rangle - \left\langle \frac{\partial \mathcal{L}}{\partial w_2}, w_2 \right\rangle = 0$, as has been previously noted by Du et al. (2018); Liang et al. (2019); Tanaka et al. (2020).

General inversion symmetry. A more general inversion symmetries can be defined by the group $\text{GL}_1(\mathbb{R})$ and action $(\theta_1, \theta_2) \mapsto (\alpha^{k_1}\theta_1, \alpha^{-k_2}\theta_2)$, which occur in networks with quadratic activation functions. A stronger form of invertible symmetry occurs for linear networks under the action of the group $GL_k(\mathbb{R})$ of $k \times k$ invertible matrices defined by $(\theta_1, \theta_2) \mapsto (G\theta_1, \theta_2 G^{-1})$, as noticed previously by Arora et al. (2018a); Du et al. (2018).

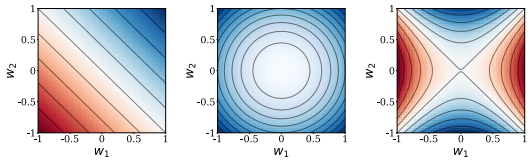
4 Conserved Quantities Constrain the Learning Dynamics

We now explore how geometric constraints on gradients and Hessians arising as a consequence of symmetry, derived above, impact network learning dynamics. In this work we focus on stochastic gradient descent (SGD), the workhorse of modern deep learning optimization. We will consider a model parameterized by θ, a training dataset $\{x_1, \ldots, x_N\}$ of size N, a training loss $\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(\theta, x_i)$ with corresponding gradient $g(\theta) = \frac{\partial \mathcal{L}}{\partial \theta}$, and an L_2 regularization constant $\lambda \geq 0$.

The gradient descent update with learning rate η is $\theta^{(n+1)} = (1 - \eta\lambda)\theta^{(n)} - \eta g(\theta^{(n)})$, which is a forward Euler discretization with step size η of the ordinary differential equation (ODE)

$$\frac{d\theta}{dt} = -g(\theta) - \lambda\theta. \tag{7}$$

In the limit as $\eta \to 0$, gradient descent exactly matches the dynamics of this ODE, which is commonly referred to as gradient flow. Equipped with a continuous model for the learning dynamics, we now ask how do the dynamics interact with the geometric properties introduced by symmetries?



(a) Translation (b) Scale (c) Inversion

Figure 2: **Visualizing conservation.** Associated with each symmetry is a conserved quantity constraining the gradient flow dynamics to a high dimensional manifold. For translation symmetry (a) the flow is constrained to a hyperplane with the intercept conserved. For scale symmetry (b) the flow is constrained to a sphere with the radius conserved. For inversion symmetry (c) the flow is constrained to high-dimensional hyperbolas with conserved axes. The color represents the value of the conserved quantity, where blue is positive and red is negative.

Symmetry leads to conservation. Consider some subset of the parameters $\theta \subseteq \Theta$ that respects either a translation, scale, or inversion symmetry. As shown in section 3, the partial gradient of the loss with respect to these parameters satisfies an equality constraint. For example, in the case of translation symmetry $\langle g(\theta), \mathbb{1} \rangle = 0$. Projecting both sides of the gradient flow learning dynamics in equation (7) onto the constant generator vector field $\mathbb{1}$ of translations then yields $\left(\frac{d}{dt} + \lambda\right)\langle \theta, \mathbb{1} \rangle = 0$. Solving the initial value problem given by this ODE results in the planar relationship

$$\langle \theta(\tau), \mathbb{1} \rangle = e^{-\lambda\tau}\langle \theta(0), \mathbb{1} \rangle. \tag{8}$$

Analogously, we can show that for scale symmetry we obtain the spherical relationship

$$|\theta(\tau)|^2 = e^{-2\lambda\tau}|\theta(0)|^2, \tag{9}$$

and for inversion symmetry we obtain the hyperbolic relationship

$$|\theta_1(\tau)|^2 - |\theta_2(\tau)|^2 = e^{-2\lambda\tau}(|\theta_1(0)|^2 - |\theta_2(0)|^2). \tag{10}$$

Each of these equations define a conserved constant of motion through training when $\lambda = 0$ and an exponentially decaying quantity when $\lambda > 0$. This result is strikingly similar to Noether's theorem

Noether, a fundamental relationship between symmetry and conservation for physical systems governed by Lagrangian dynamics.

L_2 regularization removes degrees of freedom. Despite its widespread adoption as a form of explicit regularization, the actual impact of L_2 regularization on the training dynamics of neural networks is not well understood. Some of the proposed regularizing mechanisms are increasing the effective learning rate, penalizing the input-output Jacobian norm, and reducing the damping coefficient for second-order optimization methods Zhang et al. (2018). However, a much clearer picture of L_2 regularization's impact can be found in the conservation equations (8), (9), and (10). When $\lambda > 0$ the conserved quantities exponentially decay through training at a rate proportional to the regularization constant. Geometrically, this means the degrees of freedom associated with the network symmetry, the intercepts of the hyperplane, the radius of the sphere, the axes of the hyperbola, diminish exponentially to zero.

5 A Continuous Model for Stochastic Gradient Descent

In section 4 we combined the geometric constraints introduced by symmetries with gradient flow to make exact predictions for simple combinations of parameters during training. However, there is something strange about our predictions. For example, in the case of parameters with scale symmetry, such as the weight parameters before a batch normalization layer, we would predict that SGD training with L_2 regularization should result in the norm of the weights exponentially decaying to zero, something we know is not true in practice. So what could be going wrong? Gradient flow is too simple of a continuous model for SGD training. It fails to incorporate the effect of stochasticity introduced by random batches and discretization introduced by a finite learning rate. Here, we construct a more realistic continuous model for stochastic gradient descent.

5.1 Modeling Stochasticity

Stochastic gradients $\hat{g}_{\mathcal{B}}(\theta)$ arise when we consider a minibatch $\mathcal{B}$ of size S drawn uniformly from the indices $\{1, \ldots, N\}$ forming the unbiased gradient estimate $\hat{g}_{\mathcal{B}}(\theta) = \frac{1}{S}\sum_{i \in \mathcal{B}} \ell(\theta, x_i)$. When the batch size is much smaller than the size of the dataset, $S \ll N$, then we can model the batch gradient as an average of S i.i.d. samples from a noisy version of the true gradient $g(\theta)$. Using the central limit theorem, we assume $\hat{g}_{\mathcal{B}}(\theta) - g(\theta)$ is a Gaussian random variable with mean $\mu = 0$ and covariance matrix $\Sigma = \frac{1}{S}G(\theta)G(\theta)^{\intercal}$. Under this assumption, the stochastic gradient update can be written as $\theta^{(n+1)} = (1 - \eta\lambda)\theta^{(n)} - \eta g(\theta^{(n)}) + \frac{\eta}{\sqrt{S}}G(\theta)\xi$, where ξ is a standard normal random variable. This update is an Euler-Maruyama discretization with step size η of the stochastic differential equation

$$d\theta = -g(\theta)dt - \lambda\theta dt + \sqrt{\frac{\eta}{S}}G(\theta)dW_t, \tag{11}$$

where W_t is a standard Wiener process.

The kernel of the covariance matrix. Many stochastic differential equation approximations for SGD similar to equation (11) have been derived in previous works. In order to simplify the analysis, many of these works have then made additional assumptions on the covariance matrix $\Sigma = G(\theta)G(\theta)^{\intercal}$, such as $\Sigma = H$ where H is the Hessian matrix Jastrzębski et al. (2017), $\Sigma = C$ where C is some constant matrix Mandt et al. (2015), and $\Sigma = I$ where I is the identity matrix Chaudhari & Soatto (2018). However, without *any* additional assumptions, the differential symmetries intrinsic to neural network architectures add fundamental constraints on Σ.

Again consider some subset of the parameters $\theta \subseteq \Theta$ that respects either a translation, scale, or inversion symmetry. As we showed in section 3 the partial gradients of the loss with respect to these parameters have special geometric properties, independent of the dataset. For example, in the case of translation symmetry, both the mini-batch and the full-batch gradients are orthogonal to the all-ones vector, $\langle \hat{g}_{\mathcal{B}}(\theta), \mathbb{1} \rangle = \langle g(\theta), \mathbb{1} \rangle = 0$. This implies the stochastic noise must also observe the same properties, $\langle -\frac{1}{\sqrt{S}}G(\theta)\xi, \mathbb{1} \rangle = 0$. In order for this relationship to hold for arbitrary noise ξ, this implies $G(\theta)^T\mathbb{1} = 0$. Analogously, we have for scale symmetry $G(\theta)^{\intercal}\theta = 0$ and for the inversion symmetry $G(\theta_1)^{\intercal}\theta_1 - G(\theta_2)^{\intercal}\theta_2 = 0$. In other words, the differential symmetry inherent in neural network architectures projects the noise introduced by stochastic gradients to low rank subspaces.

5.2 Modeling Discretization

While equation (11) is certainly a more realistic model for SGD than gradient flow, its major limitation is that it still assumes an infinitesimal step size when in fact the learning rate is finite. Gradient descent always moves in the direction of steepest descent on a loss function, however, due to the finite nature of the learning rate, it fails to remain on the continuous steepest descent path. The divergence between the discrete and continuous trajectories fundamentally depends on the learning rate and the curvature of the loss. It is thus natural to assume that a realistic continuous model for SGD must incorporate both these terms in a non-trivial way.

Modified gradient flow. The effect of discretization when modeling continuous dynamics is a well studied problem in the numerical analysis of partial differential equations. One tool commonly used in this setting, is modified equation analysis Warming & Hyett (1974), which asks *how can we better model discrete steps with a continuous differential equation?* Li et al. (2017); Feng et al. (2019) and more recently Barrett & Dherin (2020), applied this approach to the dynamics of stochastic gradient descent, demonstrating that the gradient descent trajectory closely follows the steepest descent path of a modified loss function $\widetilde{\mathcal{L}}$, rather than the original loss $\mathcal{L}$. Using backward error analysis they analytically calculate this modified loss as $\widetilde{\mathcal{L}} = \mathcal{L} + \frac{\eta}{4}|\nabla\mathcal{L}|^2$. Here, we apply[1] this method to modify the forcing term $F(\theta) = g(\theta) - \lambda\theta$ in equation (11), as

$$\widetilde{F}(\theta) = (1+\lambda)g(\theta) + \lambda\theta + \frac{\eta}{2}H(\theta)\left(g(\theta) + \theta\right). \tag{12}$$

Using this forcing term results in the first-order, stochastic differential equation we term *modified gradient flow*,

$$d\theta = -\widetilde{F}(\theta)dt + \sqrt{\frac{\eta}{S}}G(\theta)dW_t. \tag{13}$$

In appendix B we considers an illustrative example providing intuition for modified gradient flow.

6 Analyzing Training Dynamics through the Lens of Symmetry

As shown in section 4, each symmetry implies a conserved quantity under gradient flow with $\lambda = 0$. We now employ modified gradient flow to examine how weight decay, gradient stochasticity and finite learning rates all interact to break the conservation law of gradient flow, and we predict the dynamics of these previously conserved quantities. To do so, we project the learning dynamics given by modified gradient flow equation (13) onto generator vector fields associated with each of the symmetries while harnessing symmetry induced geometric constraints on the gradients and Hessians[2].

6.1 Planar Dynamics of Translation Symmetry

In the case of parameters with translation symmetry, the effect of discretization essentially leaves the dynamics for the constant of motion unchanged. Combining the geometric properties of graident ($\langle g, \mathbb{1}\rangle = 0$), Hessian ($H\mathbb{1} = 0$), and stochasticity ($G(\theta)^T\mathbb{1} = 0$) introduced by translation symmetry with modified gradient flow gives the following differential equation, $\left\langle \frac{d\theta}{dt} + \widetilde{F}, \mathbb{1}\right\rangle = \left(\lambda + \frac{d}{dt}\right)\langle\theta, \mathbb{1}\rangle = 0$. The solution of this differential equation is simply $\langle\theta(t), \mathbb{1}\rangle = e^{-\lambda t}\langle\theta(0), \mathbb{1}\rangle$. Notice that this is equivalent to equation (8), which we derived using gradient flow. Thus, even when considering the effect of a finite step size, the dynamics for the parameters with translation symmetry are unchanged. This is due to the absence of curvature in the constraining hyperplane geometry.

6.2 Spherical Dynamics of Scale Symmetry

In the case of parameters with scale symmetry, the effect of discretization does distort the original geometry. A finite learning rate leads to a centrifugal force that monotonically increases the previously conserved quantity ($|\theta|^2$), while weight decay acts as a centripetal force decreasing the quantity. Competition between these two forces leads to interesting dynamics with surprising consequences

[1] Consistent with Li et al. (2017) we remove all quadratic terms in η and λ.

[2] See appendix C for specific derivations.

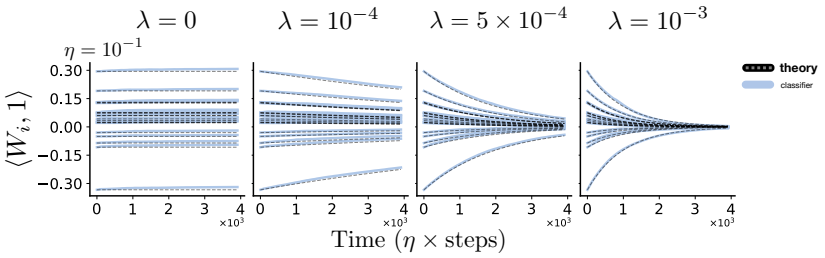


Figure 3: **The planar dynamics of VGG-16 on Tiny ImageNet.** We plot the column sum of the final linear layer of a VGG-16 model (without batch normalization) trained on Tiny ImageNet with SGD with learning rate $\eta = 0.1$, weight decay $\lambda \in \{0, 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$, and batch size $S = 256$. Colored lines are empirical column sums of the last layer through training and black dashed lines are the theoretical predictions of equation (8).

for the stationary state of these parameters. Combining the geometric constraints on the gradient ($\langle g, \theta \rangle = 0$), Hessian ($H\theta = -g$), and stochasticity ($G(\theta)^T \theta = 0$) introduced by scale symmetry with modified gradient flow gives the following differential equation,

$$\left\langle \frac{d\theta}{dt} + \widetilde{F}, \theta \right\rangle = \left(\lambda + \frac{1}{2} \frac{d}{dt} \right) |\theta|^2 - \frac{\eta}{2} |g|^2 = 0. \tag{14}$$

By solving this first order inhomogeneous differential equation, we get

$$|\theta(t)|^2 = e^{-2\lambda t} |\theta(0)|^2 + \eta \int_0^t e^{-2\lambda(t-\tau)} |g|^2 \, d\tau. \tag{15}$$

Note that the zeroth order term in η is equivalent to gradient flow equation (9), but here it is corrected by an additional non-negative integral term that is first order in η. This integral term captures the spiraling behavior of gradient descent due to the curvature of the constraining spherical geometry and a finite learning rate. We can make this exact by taking the time derivative of the equation (15) giving the temporal dynamics for the constant of motion,

$$\frac{d}{dt} |\theta(t)|^2 = -2\lambda |\theta(t)|^2 + \eta |g|^2 . \tag{16}$$

It is clear from equation (16) that even for $\lambda = 0$ the time derivative is non-negative implying the previosly conserved quantity is monotonically increasing. More generally for $\lambda \geq 0$ there is a competition between the centripetal effect of weight decay and the centrifugal effect of discretization. At the stationary state of the dynamics, these forces will balance leading the dynamics of these parameters to be constrained to the surface of a high-dimensional sphere.

Constant angular speed at the stationary state. When the dynamics of $|\theta(t)|^2$ have converged to some stationary state, then $\frac{d}{dt}|\theta(t)|^2 = 0$, which rearranging equation (16) gives the condition $\omega(t) \equiv \left|\frac{d\theta}{dt}\right| / |\theta| = \sqrt{\frac{2\lambda}{\eta}}$. Consistent with the results of Wan et al. (2020), this implies that at stationarity the angular speed $\omega(t)$ of the weights is constant and governed only by the learning rate η and weight decay constant λ.

6.3 Hyperbolic Dynamics of Inversion Symmetry

In the case of parameters with inversion symmetry, the effect of discretization also distorts the original geometry. However, unlike in the sphere, the force originating from discretization can both increase or decrease the previosly conserved quantity ($|\theta_1(t)|^2 - |\theta_2(t)|^2$). Combining the geometric properties of gradient ($\langle g_1, \theta_1 \rangle - \langle g_2, \theta_2 \rangle = 0$), Hessian ($H\theta_1 - H\theta_2 + g_{\theta_1} - g_{\theta_2} = 0$), and stochasticity ($G(\theta_1)^\intercal \theta_1 - G(\theta_2)^\intercal \theta_2 = 0$) introduced by inversion symmetry with modified gradient flow gives the following differential equation,

$$\left\langle \frac{d\theta_1}{dt} + \widetilde{F}_{\theta_1}, \theta_1 \right\rangle - \left\langle \frac{d\theta_2}{dt} + \widetilde{F}_{\theta_2}, \theta_2 \right\rangle = \left(\lambda + \frac{1}{2} \frac{d}{dt} \right) \left(|\theta_1(t)|^2 - |\theta_2(t)|^2 \right) - \frac{\eta}{2} \left(|g_{\theta_1}|^2 - |g_{\theta_2}|^2 \right) = 0. \tag{17}$$

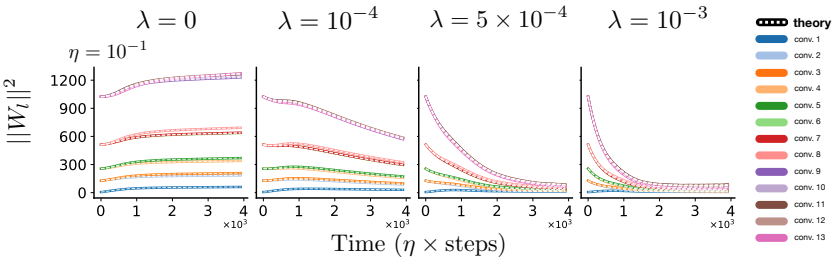


Figure 4: **The spherical dynamics of VGG-16 BN on Tiny ImageNet.** We plot the squared Euclidean norms for convolutional layers of a VGG-16 model (with batch normalization) trained on Tiny ImageNet with SGD with learning rate $\eta = 0.1$, weight decay $\lambda \in \{0, 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$, and batch size $S = 256$. Colored lines represent empirical layer-wise squared norms through training and the white dashed lines the theoretical predictions given by equation (15).

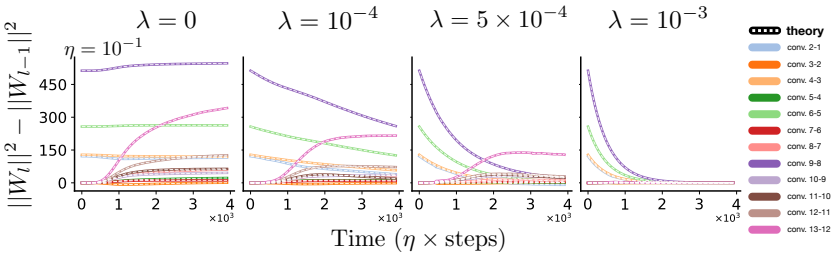


Figure 5: **The hyperbolic dynamics of VGG-16 on Tiny ImageNet.** We plot the difference between the squared Euclidean norms for consecutive convolutional layers of a VGG-16 model (without batch normalization) trained on Tiny ImageNet with SGD with learning rate $\eta = 0.1$, weight decay $\lambda \in \{0, 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$, and batch size $S = 256$. Colored lines represent empirical differences in consecutive layer-wise squared norms through training and the white dashed lines the theoretical predictions given by equation (18).

This is the same differential equation as in equation (14), just with a different forcing term. Thus, the solution for this differential equation is

$$|\theta_1(t)|^2 - |\theta_2(t)|^2 = e^{-2\lambda t}(|\theta_1(0)|^2 - |\theta_2(0)|^2) + \eta \int_0^t e^{-2\lambda(t-\tau)} \left(|g_{\theta_1}(\tau)|^2 - |g_{\theta_2}(\tau)|^2 \right) d\tau. \tag{18}$$

Again the leading order terms are equivalent to equation (10), which we derived using gradient flow, with an additional integral term. However, unlike for scale symmetry, this integral terms can be both negative or positive.

7 Conclusion

Despite being the central guiding principle in the exploration of the physical world Anderson (1972); Gross (1996), symmetry has been underutilized in understanding the mechanics of neural networks. In this paper, we developed a unifying theoretical framework that combines the power of symmetry and modified equation analysis to derive *exact* equations for combinations of parameters during training, which we experimentally verified on large scale neural networks and datasets. As shown in figures 3, 4, and 5, our theoretical predictions align remarkably well with the empirics. While, these plots are aggregated at the layer-level, our predictions are actually on combinations of parameters at the neuron-level, as shown in appendix A. Remarkably, even for models with millions of parameters, we can make exact predictions on thousands of parameter combinations through the course of training with large-scale datasets. This work provides a strong theoretical foundation for principled exploration of (i) architecture design guided by the principles of symmetry, (ii) optimization inspired by modified gradient flow, and (iii) the synergistic interaction of implicit and explicit regularization.

REFERENCES

Jing An, Jianfeng Lu, and Lexing Ying. Stochastic modified equations for the asynchronous stochastic gradient descent. *Information and Inference: A Journal of the IMA*, 2018.

Philip W Anderson. More is different. *Science*, 177(4047):393–396, 1972.

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018a.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*, 2018b.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

David GT Barrett and Benoit Dherin. Implicit gradient regularization. *arXiv preprint arXiv:2009.11162*, 2020.

Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–10. IEEE, 2018.

Vitaliy Chiley, Ilya Sharapov, Atli Kosson, Urs Koster, Ryan Reece, Sofia Samaniego de la Fuente, Vishal Subbiah, and Michael James. Online normalization for training neural networks. In *Advances in Neural Information Processing Systems*, pp. 8433–8443, 2019.

Minhyung Cho and Jaehyung Lee. Riemannian approach to batch normalization. In *Advances in Neural Information Processing Systems*, pp. 5225–5235, 2017.

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

Yuanyuan Feng, Tingran Gao, Lei Li, Jian-Guo Liu, and Yulong Lu. Uniform-in-time weak error analysis for stochastic gradient descent algorithms via diffusion approximation. *arXiv preprint arXiv:1902.00635*, 2019.

David J Gross. The role of symmetry in fundamental physics. *Proceedings of the National Academy of Sciences*, 93(25):14256–14259, 1996.

Elad Hoffer, Ron Banner, Itay Golan, and Daniel Soudry. Norm matters: efficient and accurate normalization schemes in deep networks. In *Advances in Neural Information Processing Systems*, pp. 2160–2170, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Daniel Kunin, Jonathan M Bloom, Aleksandrina Goeva, and Cotton Seed. Loss landscapes of regularized linear autoencoders. *arXiv preprint arXiv:1901.08168*, 2019.

Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pp. 2101–2110, 2017.

Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454*, 2019.

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 888–896, 2019.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Continuous-time limit of stochastic gradient descent revisited. *NIPS-2015*, 2015.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *The Journal of Machine Learning Research*, 18(1):4873–4907, 2017.

Behnam Neyshabur, Russ R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2422–2430, 2015.

E Noether. Invariante variationsprobleme, nachr. könig. gesell. wiss. göttingen, 235 (1918).

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in neural information processing systems*, pp. 901–909, 2016.

Samuel L Smith and Quoc V Le. A bayesian perspective on generalization and stochastic gradient descent. *arXiv preprint arXiv:1710.06451*, 2017.

Hidenori Tanaka, Daniel Kunin, Daniel LK Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *arXiv preprint arXiv:2006.05467*, 2020.

Twan Van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Ruosi Wan, Zhanxing Zhu, Xiangyu Zhang, and Jian Sun. Spherical motion dynamics of deep neural networks with batch normalization and weight decay. *arXiv preprint arXiv:2006.08419*, 2020.

RF Warming and BJ Hyett. The modified equation approach to the stability and accuracy analysis of finite-difference methods. *Journal of computational physics*, 14(2):159–179, 1974.

Sho Yaida. Fluctuation-dissipation relations for stochastic gradient descent. *arXiv preprint arXiv:1810.00004*, 2018.

Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*, 2018.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. *arXiv preprint arXiv:1803.00195*, 2018.