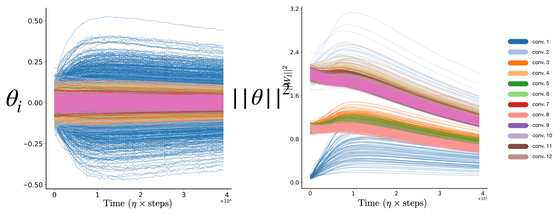
Neural Mechanics: Symmetry and Broken Conservation Laws in Deep Learning Dynamics

Daniel Kunin*, Javier Sagastuy-Brena, Surya Ganguli, Daniel L.K. Yamins, Hidenori Tanaka*†
Stanford University, †NTT Physics & Informatics Laboratories
(* equal contribution)

Background

- A better understanding of the laws governing neural network learning dynamics can have a profound impact on the optimization of artificial neural networks.
- Q. What, if anything, can we quantitatively predict about the complex learning dynamics of state-of-the-art deep learning models driven by real-world datasets?
- Existing works have made major simplifying assumptions on the network, such as restricting to identity activation functions, infinite width layers, or single hidden layers.
- Rather than introducing unrealistic assumptions, we uncover restricted, but meaningful, combinations of parameters with simplified dynamics that can be solved exactly without introducing a single assumption
- To find the parameter combinations, 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.



While the parameter dynamics (left) are noisy and chaotic, the neuron dynamics (right) are smooth and patterned.

Symmetry Leads to Conservation Laws Under Gradient Flow

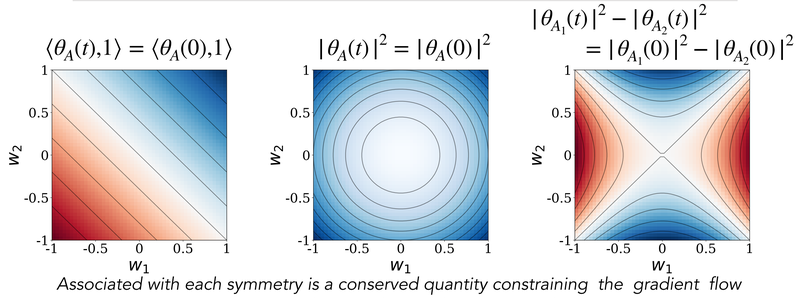
The gradient descent update with learning rate η is $\theta^{(n+1)} = \theta^{(n)} - \eta g(\theta^{(n)})$, which is a forward Euler discretization of the ODE known as gradient flow:

$$\frac{d\theta}{dt} = -g(\theta)$$

How do these learning dynamics interact with the geometric properties introduced by symmetry?

Noether's Theorem for Neural Networks: Every symmetry of a network architecture has a corresponding conserved quantity through training under gradient flow. Projecting the gradient flow dynamics onto the generator vector field generates an ODE, whose solution is a conservation law.

$$\left\langle \frac{d\theta}{dt}, \partial_\alpha \psi \right\rangle = 0$$



Associated with each symmetry is a conserved quantity constraining the gradient flow dynamics to a high-dimensional hyperplane, sphere, or hyperbola.

Combining Symmetry and Modified Flow to Derive Learning Dynamics

We now study how weight decay, momentum, stochastic gradients, and finite learning rates all interact to break these conservation laws. To do this we:

1. Consider a realistic continuous model for SGD, an equation of learning.
2. Project the learning dynamics onto the generator vector fields associated with a symmetry.
3. Harness the geometric constraints introduced by symmetry to derive simplified ODEs.
4. Solve these ODEs to obtain exact dynamics for the previously conserved quantities.

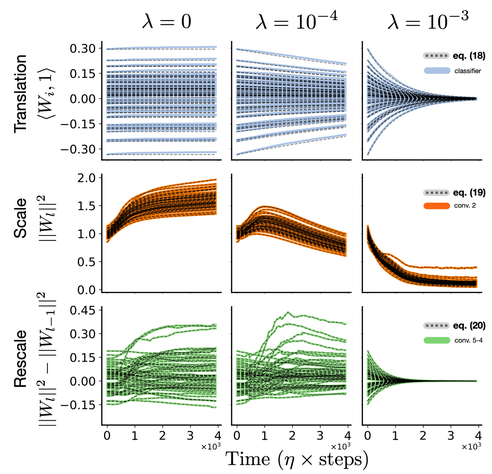
As a result, we get exact learning dynamics for combinations of parameters,

Translation: $\langle \theta_{\mathcal{A}}(t), \mathbb{1} \rangle = e^{-\lambda t} \langle \theta_{\mathcal{A}}(0), \mathbb{1} \rangle$

Scale: $|\theta_{\mathcal{A}}(t)|^2 = e^{-2\lambda t} |\theta_{\mathcal{A}}(0)|^2 + \eta \int_0^t e^{-2\lambda(t-\tau)} |g_{\mathcal{A}}|^2 \, d\tau$

Rescale: $|\theta_{\mathcal{A}_1}(t)|^2 - |\theta_{\mathcal{A}_2}(t)|^2 =$

$$e^{-2\lambda t}(|\theta_{\mathcal{A}_1}(0)|^2 - |\theta_{\mathcal{A}_2}(0)|^2) + \eta \int_0^t e^{-2\lambda(t-\tau)} \left(|g_{\theta_{\mathcal{A}_1}}|^2 - |g_{\theta_{\mathcal{A}_2}}|^2 \right) d\tau$$



These theoretical equations (dotted lines) match the empirics (colored lines) on VGG-16 trained on Tiny ImageNet with SGD.

Symmetry in the Loss Constrain 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 introduced by symmetries.

Symmetry: Given an action on the parameter vector, $\theta \mapsto \psi(\theta, \alpha)$ a function possesses a differentiable symmetry if, $f(\psi(\theta, \alpha)) = f(\theta)$ for (θ, α).

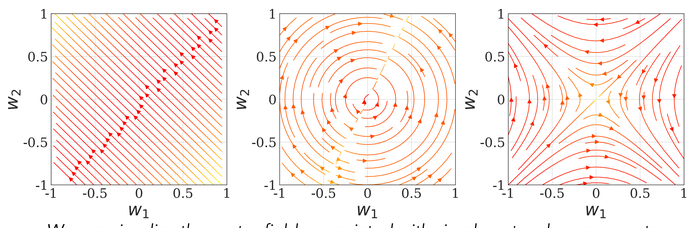
Geometric constraints: By evaluating gradient and Hessian at identity, we get,

Gradient constraint ($g = \partial_\theta f$): $\langle \nabla f, \partial_\alpha \psi \rangle = 0$

Hessian constraint, ($H = \partial^2_\theta f$): $\mathbf{H} f \partial_\theta \psi \partial_\alpha \psi + \partial_\theta \partial_\alpha \psi \nabla f = 0$

We consider the specific setting of a neural network parameterized by θ, the training loss $\mathscr{L}(\theta)$, and three families of symmetries that commonly appear in modern neural network architectures.

	Translation	Scale	Rescale
Symmetry	$\mathscr{L}(\theta) = \mathscr{L}(\theta + \alpha 1)$	$\mathscr{L}(\theta) = \mathscr{L}(\alpha\theta)$	$\mathscr{L}(\theta_{A1}, \theta_{A2}) = \mathscr{L}(\alpha\theta_{A1}, \alpha^{-1}\theta_{A2})$
Gradient	$\langle g, 1 \rangle = 0$	$\langle g, \theta \rangle = 0$	$\langle g, \theta_{A_1} - \theta_{A_2} \rangle = 0$
Hessian	$\langle H, 1 \rangle = 0$	$\langle H, \theta \rangle = -g$	$H(\theta_{A_1} - \theta_{A_2}) + g_{A1} - g_{A2} = 0$



We can visualize the vector fields associated with simple network components that have translation, scale, and rescale symmetry in 2D.

A Realistic Continuous Model for Stochastic Gradient Descent

Gradient flow is too simple of a continuous model for realistic SGD training.

Modeling weight decay and momentum

Explicit regularization with constant λ and momentum with constant β can be modeled through simple modifications to gradient flow,

$$(1 - \beta)\frac{d\theta}{dt} = -g(\theta) - \lambda\theta$$

Modeling stochasticity

Stochastic gradients arise when we consider a random batch of size S forming an unbiased gradient estimate. We can model the batch gradient as a Gaussian random variable giving a stochastic differential equation,

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

Modeling discretization

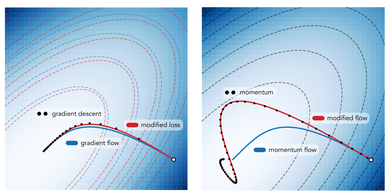
Gradient descent always moves in the direction of steepest descent, however, due to the finite learning rate, it fails to remain on the continuous steepest descent path, which can be modeled with modified equation analysis.

Modified Loss - Introduces higher order derivatives of the loss, effectively modifying the loss landscape itself.

$$\frac{d\theta}{dt} = -g(\theta) - \frac{\eta}{2} H(\theta) g(\theta)$$

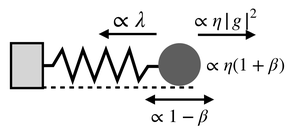
Modified Flow - Introduces higher order temporal derivatives modifying the flow directly.

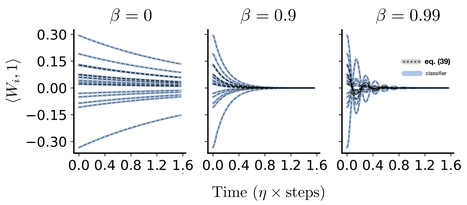
$$\frac{d\theta}{dt} = -g(\theta) - \frac{\eta}{2} \frac{d^2\theta}{dt^2}$$



We can visualize how modified loss (left) and modified flow (right) accurately model the effect of discrete SGD steps in a quadric setting.

Harmonic oscillation with momentum. When considering the learning dynamics of momentum, the solutions we obtain take the form of driven harmonic oscillators, where the optimization hyperparameters have physical interpretations.





We plot the column sum of the final linear layer of a VGG-16 model (without batch normalization) trained on Tiny ImageNet.

Conclusion and Future Work

- We constructed a unifying theoretical framework harnessing the geometric properties of symmetry and realistic continuous equations for learning that model weight decay, momentum, stochasticity, and discretization.
- This work provides a first step towards understanding the mechanics of learning in neural networks without unrealistic simplifying assumptions.
- In future studies, we'd like to better optimize neural networks by harnessing the understanding of how realistic optimizer breaks previously conserved parameter combinations to adaptively control trajectory of training.