ReFoRM Reading Group

Rethinking Foundations Real-World ML

Anay Mehrotra, Amin Saberi, Grigoris Velegkas

Warm-started from slides by Andrew Ilyas and Amin Saberi

Welcome to ReFoRM!

What is this reading group about? Foundations of *“real-world”* ML

Welcome to ReFoRM!

What is this reading group about? Foundations of *"real-world"* ML

How is "real-world" ML different from "idealized" ML?

Idealized picture: $$\theta^\star = \arg\min_{\theta \in \Theta} \mathbb{E}_{z \sim D}[\ell(z_i; \theta)]$$

Welcome to ReFoRM!

What is this reading group about? Foundations of *"real-world"* ML

How is "real-world" ML different from "idealized" ML?

Idealized picture: $$\theta^\star = \arg\min_{\theta \in \Theta} \mathbb{E}_{z \sim D}[\ell(z_i; \theta)]$$

Decisions:

- How to choose the parameter space to avoid *overfitting*?
- What (convex) *loss function ℓ* to choose?
- Which *optimization algorithm* to use?

Welcome to ReFoRM!

What is this reading group about? Foundations of *“real-world”* ML

How is “real-world” ML different from “idealized” ML?

Idealized picture: $$\theta^\star = \arg\min_{\theta \in \Theta} \mathbb{E}_{z \sim D}[\ell(z_i; \theta)]$$

Decisions:

- How to choose the parameter space to avoid *overfitting*?
- What (convex) *loss function* ℓ to choose?
- Which *optimization algorithm* to use?

Guarantees: Convergence rates, generalization bounds, uncertainty quantification (via confidence intervals), performance on different distributions,...

Welcome to ReFoRM!

What is this reading group about? Foundations of "real-world" ML

How is "real-world" ML different from "idealized" ML?

Real-world ML: $$\theta^\star = \arg\min_{\theta \in \Theta} \mathbb{E}_{z \sim D}[\ell(z_i; \theta)]$$

Messy Dataset D

+

Very Expressive Hypothesis Classes Θ

Welcome to ReFoRM!

What is this reading group about? Foundations of "real-world" ML

How is "real-world" ML different from "idealized" ML?

Real-world ML: $$\theta^\star = \arg\min_{\theta \in \Theta} \mathbb{E}_{z \sim D}[\ell(z_i; \theta)]$$

Messy Dataset D + *Very Expressive Hypothesis Classes* Θ $\longrightarrow$ *Learning Algorithm (min)*

Learning Algorithm (min) $\uparrow$ *Regularizer + Optimizer*

Welcome to ReFoRM!

What is this reading group about? Foundations of "real-world" ML

How is "real-world" ML different from "idealized" ML?

Real-world ML: $$\theta^\star = \arg\min_{\theta \in \Theta} \mathbb{E}_{z \sim D}[\ell(z_i; \theta)]$$

$$\begin{array}{ccccc}
\textit{Messy Dataset } D + \textit{Very Expressive Hypothesis Classes } \Theta & \longrightarrow & \textit{Learning Algorithm (min)} & \longrightarrow & \textit{Base Model } \theta^\star \\
 & & \uparrow & & \downarrow \\
 & & \textit{Regularizer + Optimizer} & & \textit{Post-Trained Model } \widetilde{\theta}^\star
\end{array}$$

Welcome to ReFoRM!

What is this reading group about? Foundations of “real-world” ML

How is “real-world” ML different from “idealized” ML?

Real-world ML: $$\theta^\star = \arg\min_{\theta \in \Theta} \mathbb{E}_{z \sim D}[\ell(z_i; \theta)]$$

$$\begin{tikzcd}
\textit{Messy Dataset } D + \textit{Very Expressive Hypothesis Classes } \Theta \arrow[r] & \textit{Learning Algorithm (min)} \arrow[r] & \textit{Base Model } \theta^\star \arrow[d] \\
& \textit{Regularizer + Optimizer} \arrow[u] & \textit{Post-Trained Model } \widetilde{\theta}^\star
\end{tikzcd}$$

Implications: Unpredictability, theoretical wisdom might not apply, new considerations, need to understand new phenomena, …

Goal of this group

What do rigorous foundations for this new age of ML look like?

How can tools from statistics, CS theory, and operations inform a *better understanding* of machine learning algorithms and systems?

What are the right questions to ask, and phenomena to explain—at what *level of abstraction* should we be aiming to explain them?

What theoretical models not only *explain* unexpected phenomena, but also *predict* new phenomena that we can verify experimentally?

Intended format (thanks for signing up!)

Goal: Build intuition, leverage group's diversity, start collaborations (bringing new perspectives from everyone's field)

Sign up: https://tinyurl.com/reform-ml-signup-w26

Goal(*s*) of the discussant (1-2 every week):

1. A single "deep dive" per week about one subject (can be multiple papers)
2. We have suggested several papers for each week, *more* than one can cover thoroughly in a week. Pick a small + focused paper set and read thoroughly
3. Prepare a 20-30 minute presentation, accessible to a second year PhD student, focusing on (a) *seeding discussion* and (b) *identifying gaps and connections,* and (c) *formulating open problems*

Everyone else: Read paper/watch talk/something! *Try to come with some familiarity*

Introductions!

What is your *name*?

What *program and year* are you in?

What *focus area* are you most interested in?

What are you *working* on? What do you *want to work* on?

What brought you to this reading group?

Outline for the Quarter

Introduce the theme for this quarter: *Training dynamics and optimization*

The quarter is divided into three *sessions* (each two-week long)

Each Session's Goal: *Explore a sub-area in depth*

Understand the known results

Identify gaps

Formulate open problems

Sessions This Quarter

A) **Sharpness and Training Dynamics** (Jan 22, Feb 5)

B) **Overfitting and Generalization** (Feb 12, Feb 19)

C) **Grokking and Emergent Abilities** (Feb 26, Mar 5)

Meetings This Quarter

January 22nd (today!)	Introduction to edge of stability (EoS)
January 29th	Skipping due to ICML deadline
February 5th	Explanations of EoS / Sharpness & Generalization
February 12th	Double Descent
February 19th	Benign Overfitting
February 26th	Grokking
March 5th	Other emergent abilities
March 12th	Reserved for extra meeting on above / different topic

Session 1
Edge *of* Stability

GRADIENT DESCENT ON NEURAL NETWORKS TYPICALLY OCCURS AT THE EDGE OF STABILITY

Jeremy Cohen   Simran Kaur   Yuanzhi Li   J. Zico Kolter[1]   and   Ameet Talwalkar[2]
Carnegie Mellon University and: [1]Bosch AI   [2] Determined AI
Correspondence to: `jeremycohen@cmu.edu`

Why? The State of ML Optimization

The state of neural network optimization, today:

1. Many *optimization algorithms* (SGD, momentum, Adam, muon, …), can *successfully train* neural networks (CNNs, transformers, …)

2. In *simplified settings* (quadratic and convex functions), we *understand* what these algorithms do, and why they succeed

3. However, we *do not understand* how they function in *realistic settings*

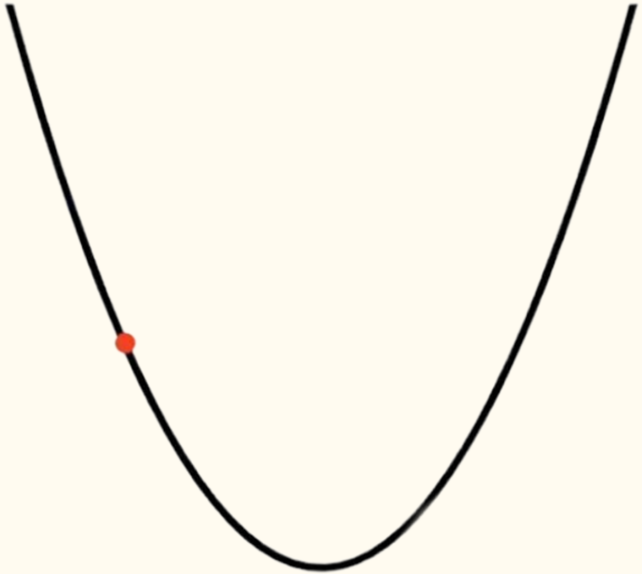
> **Q:** Can we use principled empirical observations to develop an understanding? *Even for the simplest optimizer – gradient descent?*

GD and Sharpness with Quadratic Functions

Consider running gradient descent with step size η on a 1-dimensional quadratic

The behavior depends on the relationship between the *step size* η and *curvature* a

- If $a < 2/\eta$, gradient descent *converges*
- If $a > 2/\eta$, gradient descent *diverges*



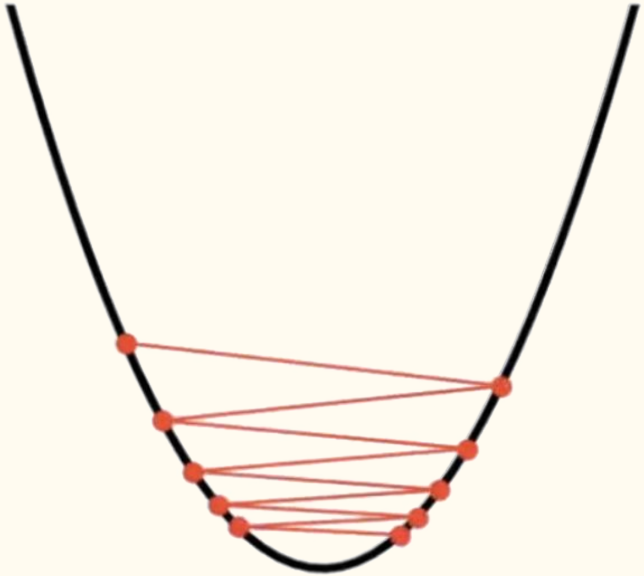
$$f(x) = \frac{1}{2}ax^2 + bx + c$$

GD and Sharpness with Quadratic Functions

Consider running gradient descent with step size η on a 1-dimensional quadratic

The behavior depends on the relationship between the *step size* η and *curvature* a

- If $a < 2/\eta$, gradient descent *converges*
- If $a > 2/\eta$, gradient descent *diverges*



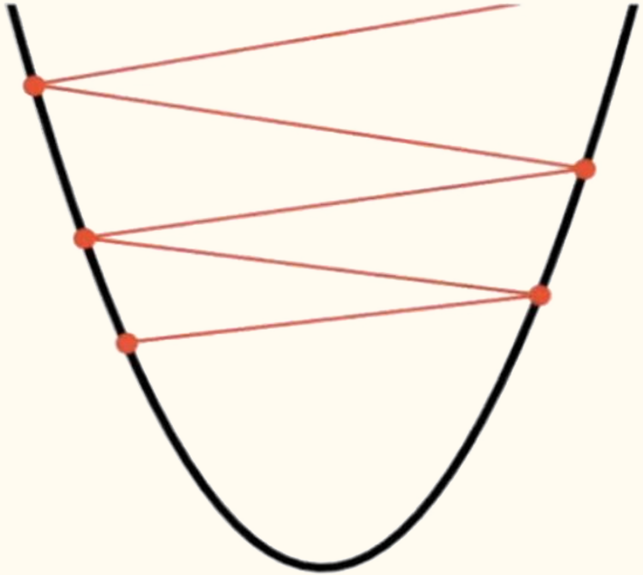
$$f(x) = \frac{1}{2}ax^2 + bx + c$$

GD and Sharpness with Quadratic Functions

Consider running gradient descent with step size η on a 1-dimensional quadratic

The behavior depends on the relationship between the *step size* η and *curvature* a

- If $a < 2/\eta$, gradient descent *converges*
- If $a > 2/\eta$, gradient descent *diverges*



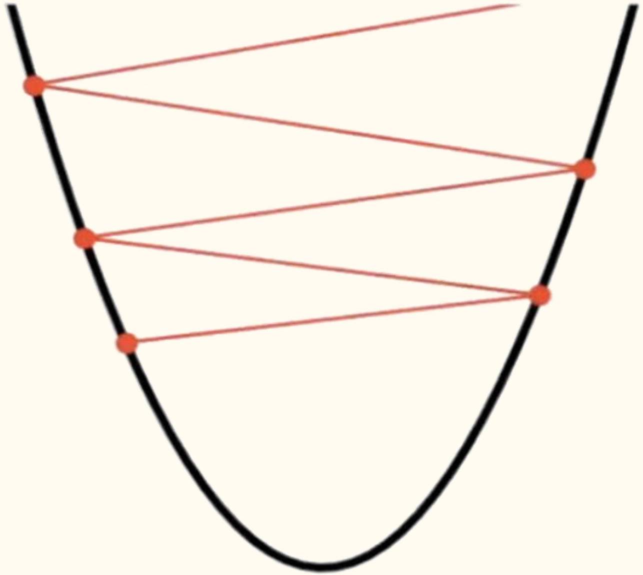
$$f(x) = \frac{1}{2}ax^2 + bx + c$$

GD and Sharpness with Quadratic Functions

Consider running gradient descent with step size η on a 1-dimensional quadratic

The behavior depends on the relationship between the *step size* η and *curvature* a

- If $a < 2/\eta$, gradient descent *converges*
- If $a > 2/\eta$, gradient descent *diverges*



Sharpness at x: $|\nabla^2 f(x)|$

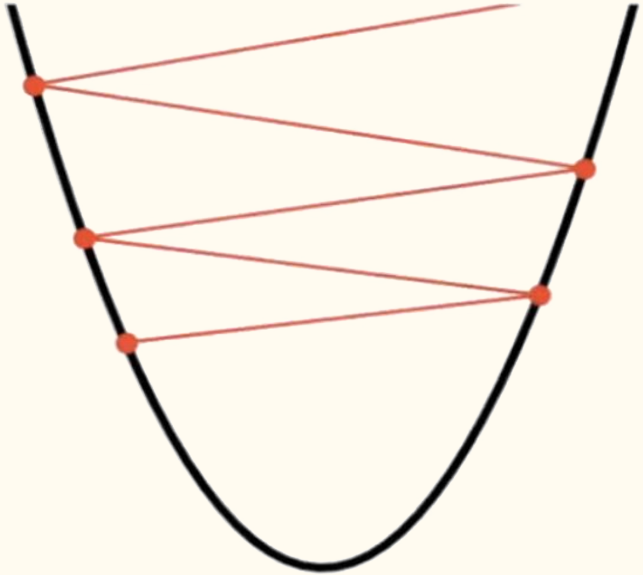
$$f(x) = \frac{1}{2}ax^2 + bx + c$$

GD and Sharpness with Quadratic Functions

Consider running gradient descent with step size η on a 1-dimensional quadratic

The behavior depends on the relationship between the *step size* η and *curvature* a

- If $a < 2/\eta$, gradient descent *converges*
- If $a > 2/\eta$, gradient descent *diverges*



Natural generalization to higher-dimensions

$$\textbf{Sharpness at } x\textbf{:} \quad \|\nabla^2 f(x)\|_2$$

$$f(x) = \frac{1}{2}ax^2 + bx + c$$

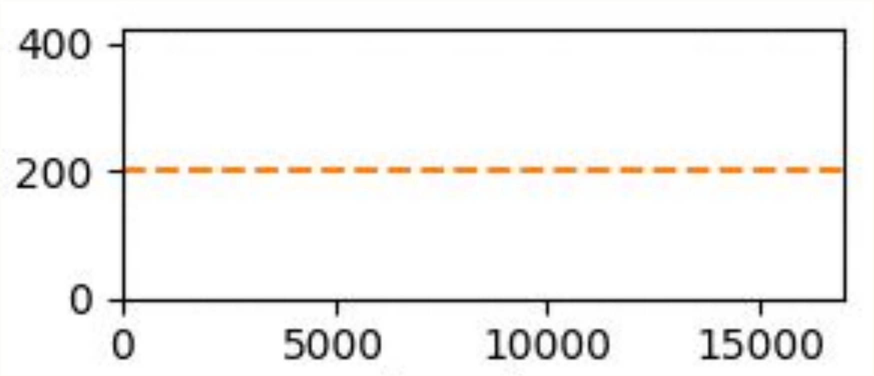
Sharpness in deep learning

Sharpness: Maximum eigenvalue of Hessian of training loss *f(x)*

How does sharpness behave in neural network training?

Rough Observation: Initially, the sharpness increases until it reaches a stable value after which it stabilizes

sharpness



Sharpness in deep learning

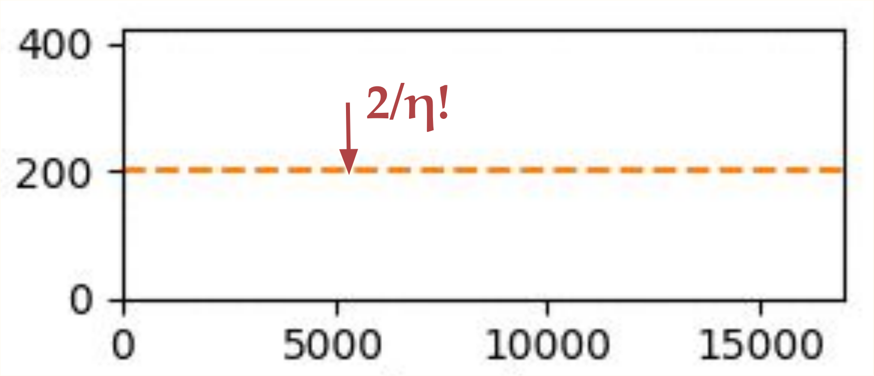
Sharpness: Maximum eigenvalue of Hessian of training loss $f(x)$

How does sharpness behave in neural network training?

Observation 1 (Progressive sharpening):
If sharpness is less than $2/\eta$ (i.e., gradient descent is stable), *sharpness tends to increase*

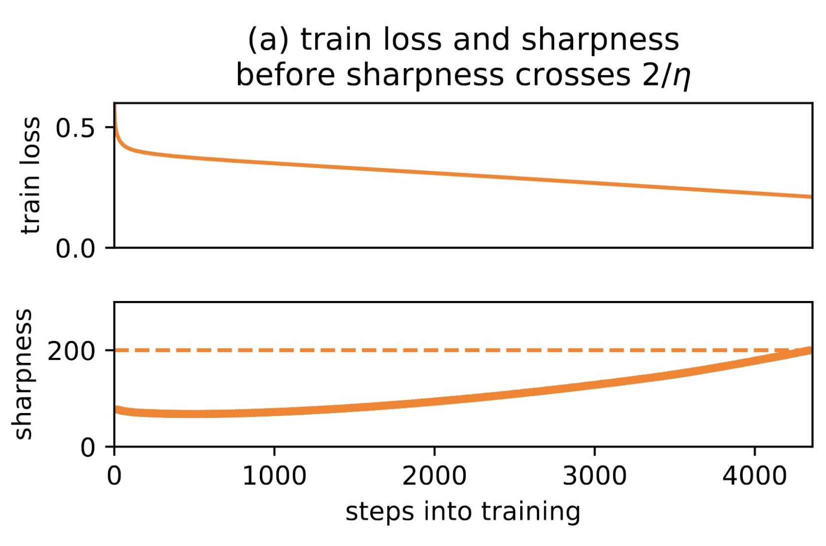
Observation 2 (Edge of stability):
After this, sharpness hovers *just above* $2/\eta$ for the remainder of training

sharpness

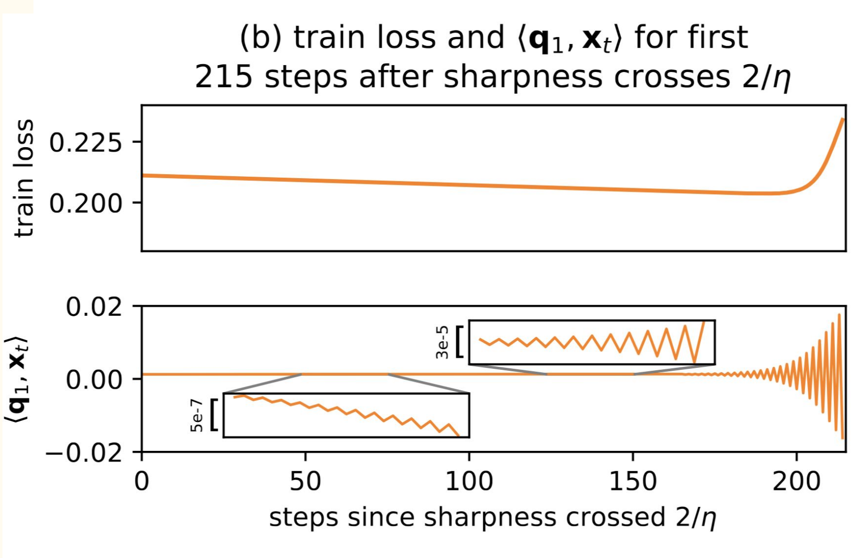


The network is a fully-connected architecture with two hidden layers of width 200, and tanh activations.

Sharpness and Train Loss



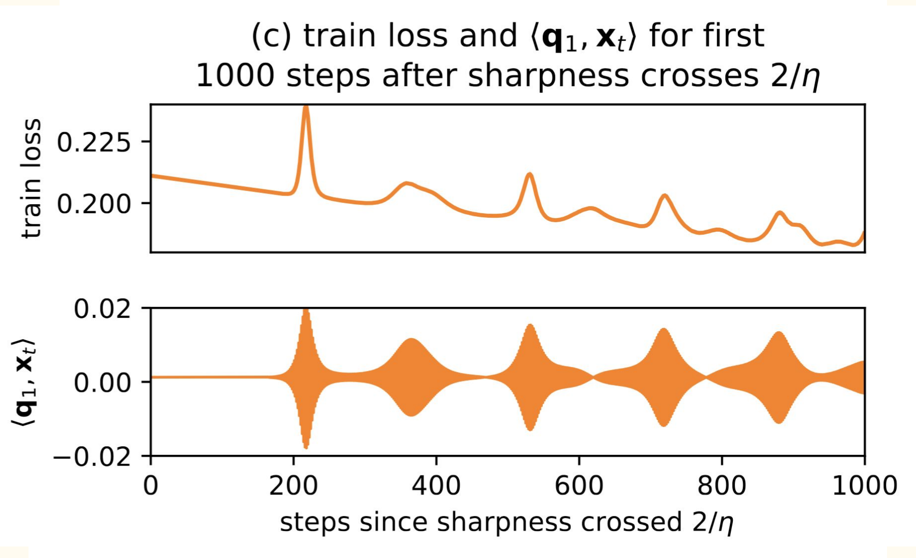
Sharpness and Train Loss



Unclear what happens next:

1. If loss is quadratic, GD *diverges*
2. GD might "jump" to a flatter region and *train loss stagnates*
3. GD might *not escape* local minima

Sharpness and Train Loss

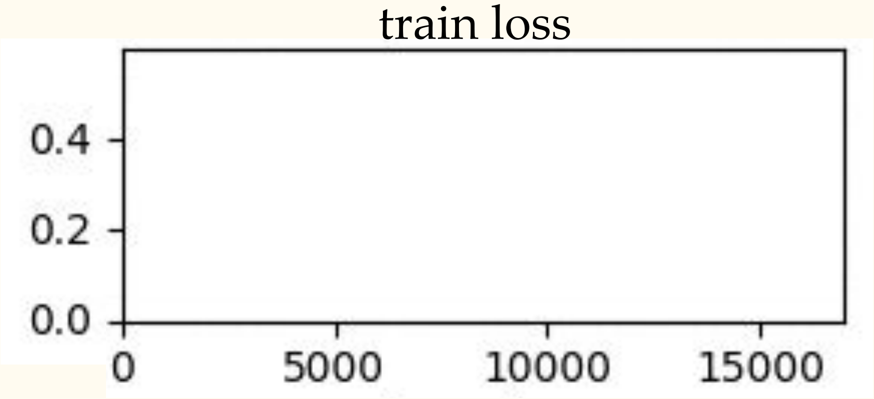


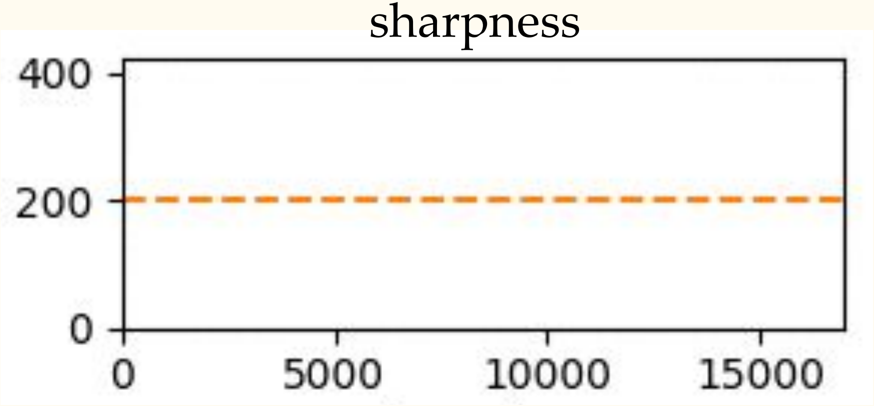
Unclear what happens next:

1. If loss is quadratic, GD *diverges*
2. GD might "jump" to a flatter region and *train loss stagnates*
3. GD might *not escape* local minima

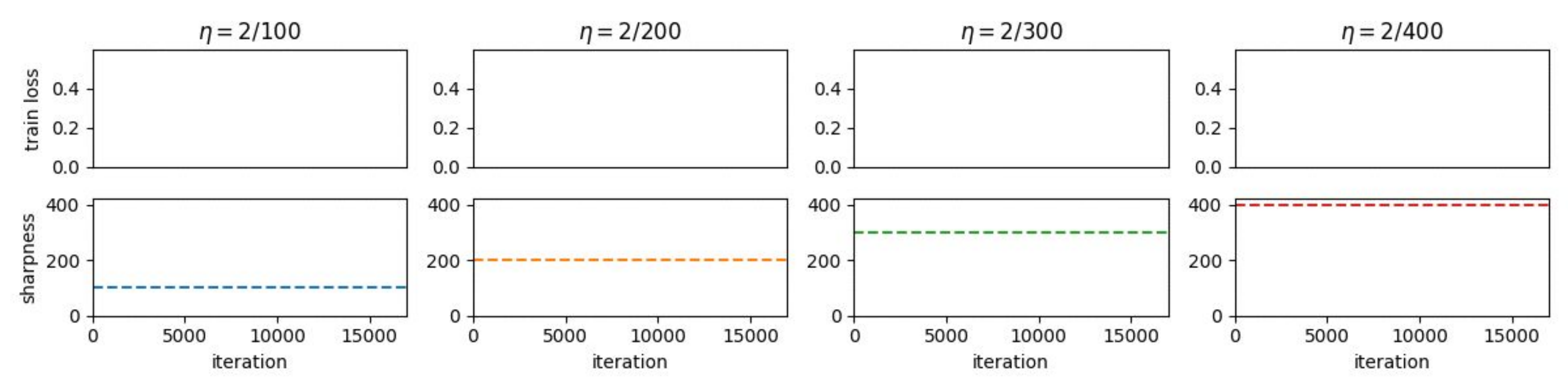
None of these happen! GD makes progress and training loss reduces

Updated Picture



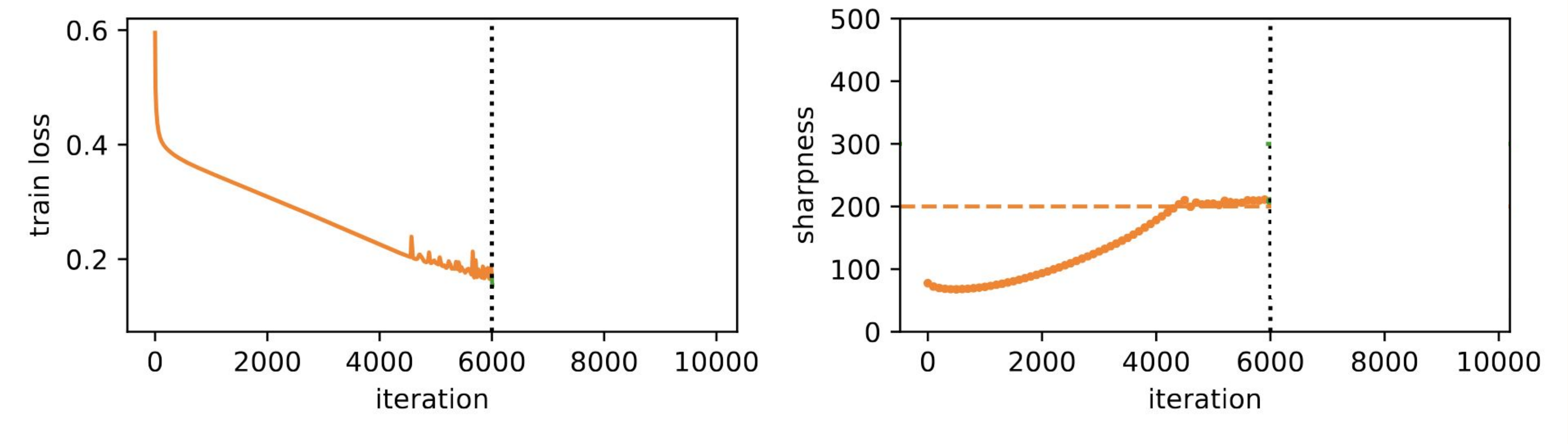


Different Step-Sizes η



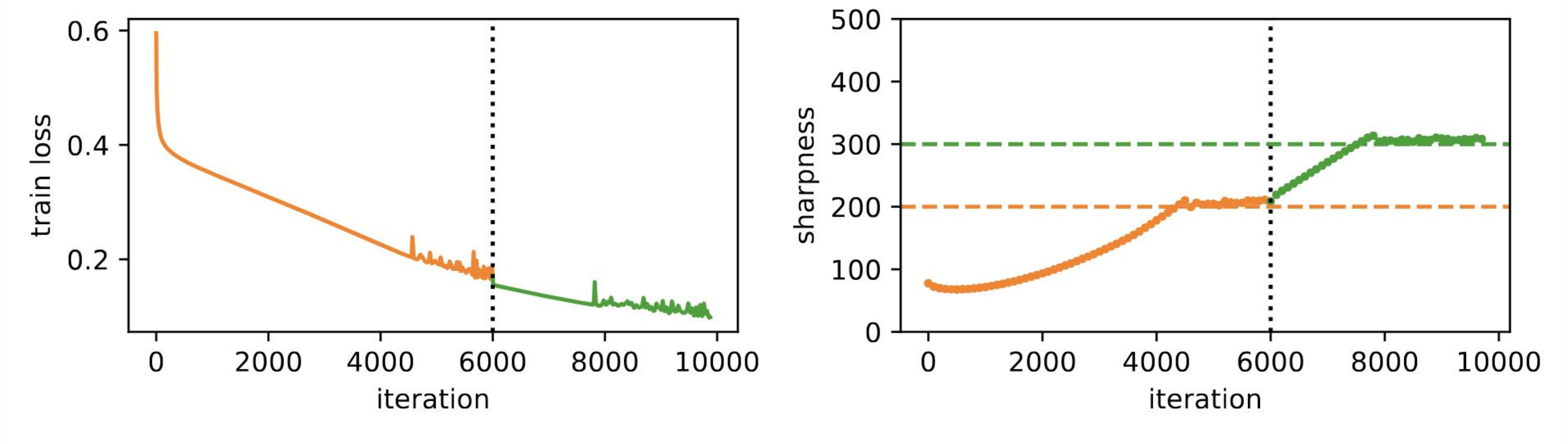
Different Step-Sizes in the Same Run

Drop η from 2/200 to 2/300 at iteration 6000



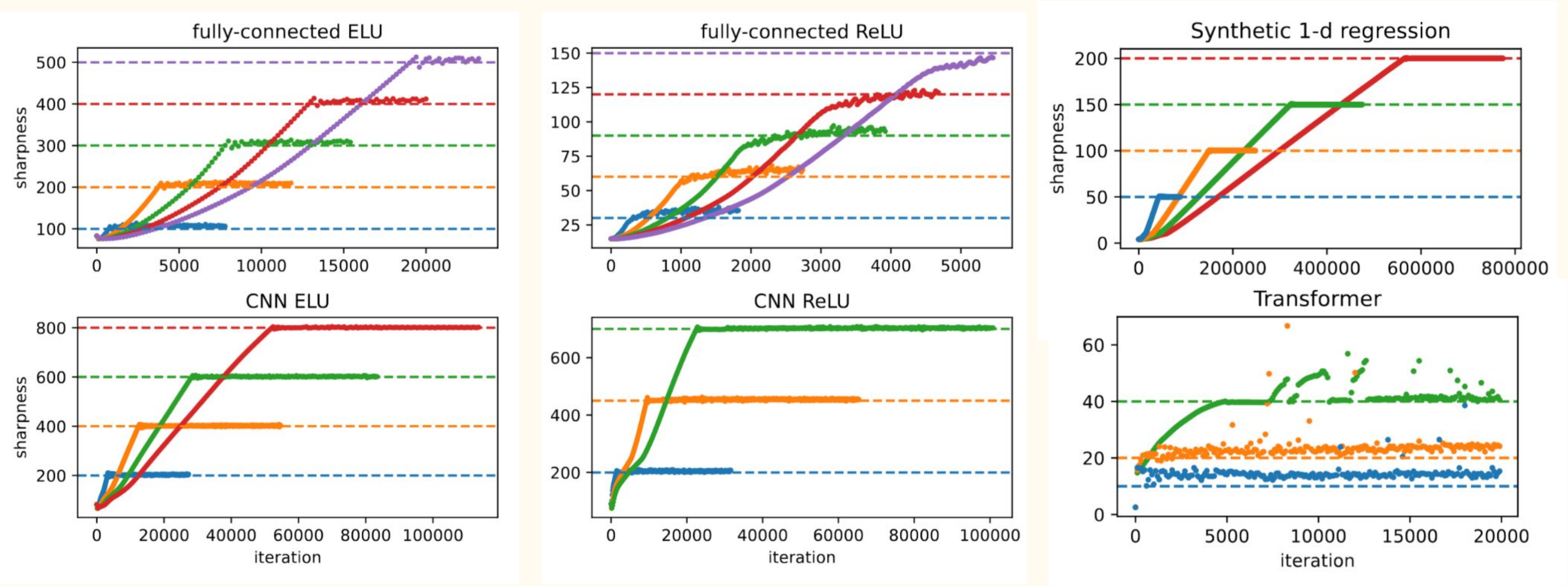
Different Step-Sizes in the Same Run

Drop η from 2/200 to 2/300 at iteration 6000

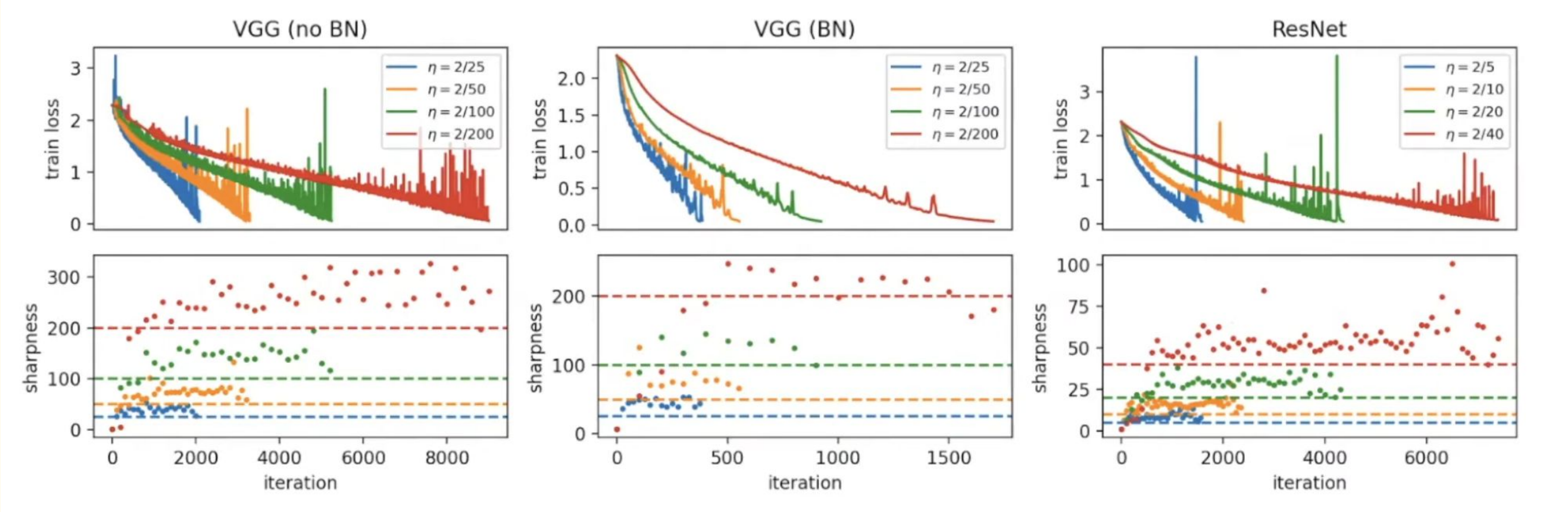


On changing η, GD *re-enters* the progressive sharpness regime

Different Tasks and Architectures



Different Tasks and Architectures

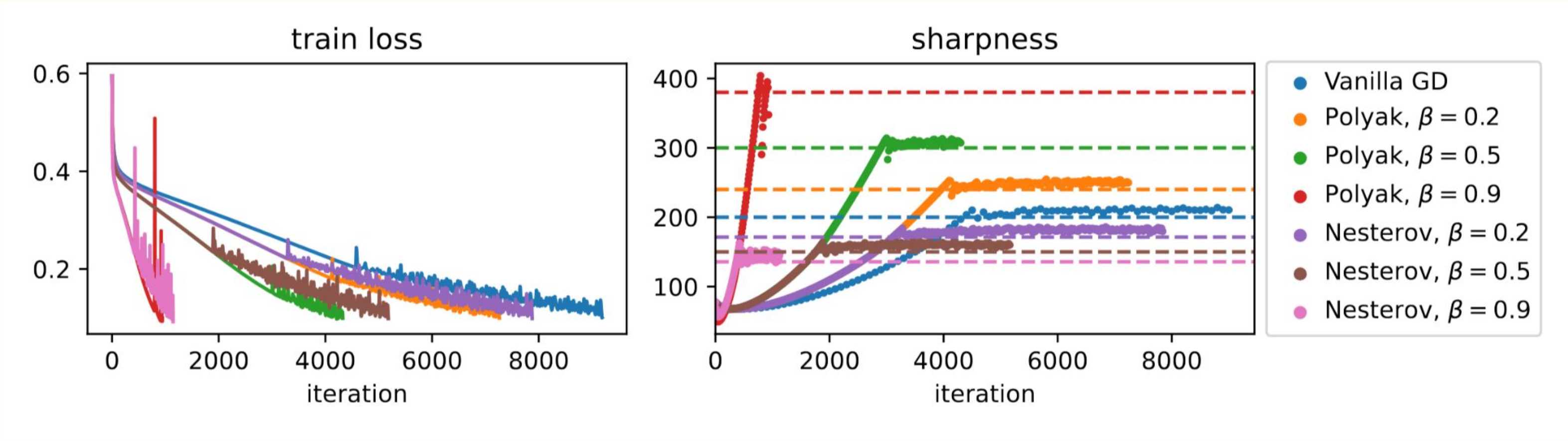


Step-size η used to train is 50x too large to observe progressive sharpening

For (Some) Other Optimizers

Polyak Momentum: $v_{t+1} = \beta v_t - \eta \nabla f(x_t), \quad x_{t+1} = x_t + v_{t+1}$

Nesterov Momentum: $v_{t+1} = \beta v_t - \eta \nabla f(x_t + \beta v_t), \quad x_{t+1} = x_t + v_{t+1}$



Q: *Other optimizers?* It does *not* apply to SGD unless batch size is large…

Implications for optimization theory

The behavior of gradient descent at the Edge of Stability casts doubt on traditional step-size choice — selected based on quadratic approx

Perhaps one should consider higher-order Taylor approximations?

Self-Stabilization: The Implicit Bias of Gradient Descent at the Edge of Stability

Alex Damian*
Princeton University
ad27@princeton.edu

Eshaan Nichani*
Princeton University
eshnich@princeton.edu

Jason D. Lee
Princeton University
jasonlee@princeton.edu

Understanding Optimization in Deep Learning with Central Flows

Jeremy Cohen*
Carnegie Mellon and Flatiron Institute
jmcohen.github.io

Alex Damian*
Princeton University
alex-damian.github.io

Ameet Talwalkar
Carnegie Mellon University

J. Zico Kolter
Carnegie Mellon University

Jason D. Lee
Princeton University

Several Caveats

- *Loss Function Choice:* With cross-entropy loss, the sharpness often drops at end of training
- *Architecture + Dataset:* For shallow/wide networks, or simple datasets, sharpness does not rise to $2/\eta$
- *Batch normalization:* Need to look at sharpness between iterates
- *Non-differentiable components:* instability sometimes begins when the sharpness is a bit less than $2/\eta$

Open Questions

1. *Connection to generalization:* Does the EoS regime impart inductive biases that help finding “flatter” (often also more generalizable) minima?

Open Questions

1. *Connection to generalization:* Does the EoS regime impart inductive biases that help finding "flatter" (often also more generalizable) minima?
2. *Extension to "Fancier" Optimizers:* Does EOS arise for fancier optimizers (muon, SOAP, Shampoo, ...)? If yes, can this be used to understand and *improve* their performance?

Open Questions

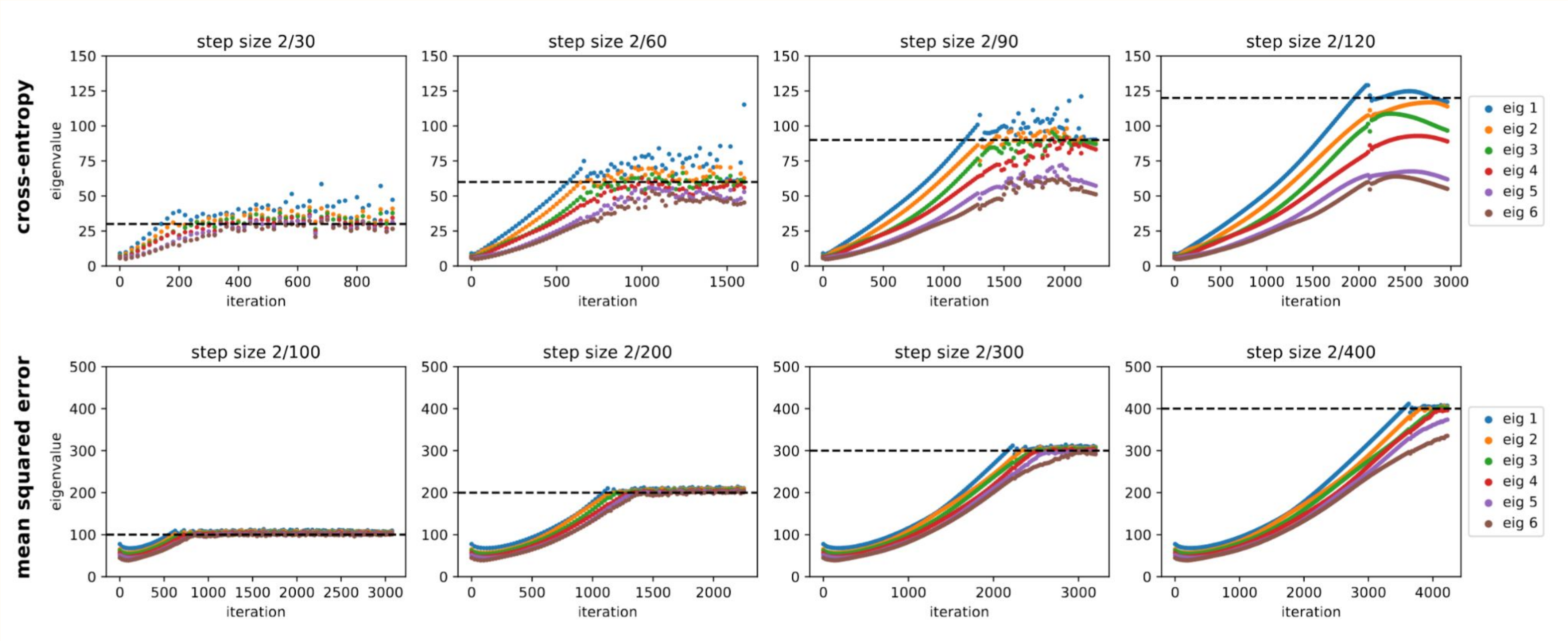
1. *Connection to generalization:* Does the EoS regime impart inductive biases that help finding "flatter" (often also more generalizable) minima?
2. *Extension to "Fancier" Optimizers:* Does EOS arise for fancier optimizers (muon, SOAP, Shampoo, …)? If yes, can this be used to understand and *improve* their performance?
3. *Edge of Stability for SGD:* What is the correct EOS analogue for mini-batch SGD? Ongoing work…

Edge of Stochastic Stability:
Revisiting the Edge of Stability for SGD

Arseniy Andreyev* Pierfrancesco Beneventano*

Appendix

Next Few Eigenvalues



Caveat: The observation does *not* apply to SGD unless batch size is large