Multiplicative noise and heavy tails in stochastic optimization and machine learning

Michael W. Mahoney
ICSI, LBNL, and UC Berkeley

Joint work with Liam Hodgkinson and others.

Outline

Heavy-tailed Self-regularization Theory

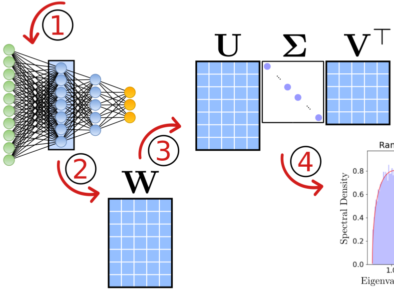
"Multiplicative noise and heavy tails in stochastic optimization," HM, ICML 2021

"Generalization Properties of Stochastic Optimizers via Trajectory Analysis," HSKM, ICML 2022

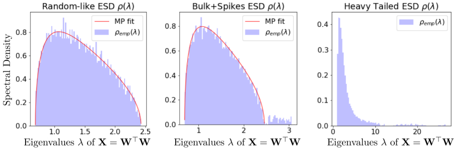
When are ensembles really effective?

What do SOTA ML models "look like"?

Analyzing DNN Weight matrices with WeightWatcher



1. Take a model
↳2. Take a weight matrix
↳3. Do Spectral analysis
↳4. Histogram of eigenvalues



- Analyze one layer of pre-trained model
- Compare multiple layers of pre-trained model
- Monitor NN properties as you train your own model

"pip install weightwatcher"

Implicit Self-Regularization in Deep Neural Networks: Evidence from RMT and Implications for Learning JMLR21

Outline

Heavy-tailed Self-regularization Theory

"Multiplicative noise and heavy tails in stochastic optimization," HM, ICML 2021

"Generalization Properties of Stochastic Optimizers via Trajectory Analysis," HSKM, ICML 2022

When are ensembles really effective?

Stochastic optimization

is the process of minimizing an objective function via the simulation of random elements.

"the backbone of modern machine learning"

Stochastic optimizers

In deep learning...

- Stochastic gradient descent (SGD)

$$w_{k+1} = w_k - \frac{\gamma}{|\Omega_k|} \sum_{i \in \Omega_k} \nabla f_i(w_k)$$

- Momentum
- Stochastic Newton methods
- Adam
- and *many* others...

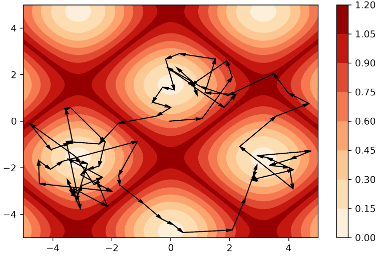
Based on classical (convex) optimization algorithms.

Stochastic component (minibatches) can allow them to work well in unconstrained *non-convex* settings.

Robbins, H., Monro, S. (1951) A stochastic approximation method. The Annals of Mathematical Statistics, pp.400—407

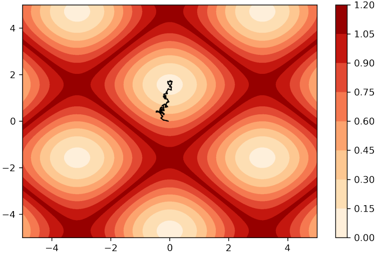
Phases of Training

Exploration
large learning rate



(sampler)

Exploitation
small learning rate



(optimizer)

Mandt, S., Hoffman, M., Blei, D. (2016) A variational analysis of stochastic gradient algorithms. ICML 2016, pp. 354–363.

A distributional approach

Investigate how a stochastic optimizer explores the loss landscape

1. Model stochastic optimization as a random dynamical system (Markov)
2. Fix all hyperparameters to particular values (time-homogeneous; no annealing)
3. Examine properties of the **stationary (invariant) distribution**

- Avoid continuous-time approximations

Our Findings

Multiplicative noise results in heavy-tailed stationary behaviour

- Tails of the stationary distribution are an indication of capacity to explore
- Decay rates in the tails that are slower than exponential are **heavy**, e.g.

$$\mathbb{P}(W > t) \approx ct^{-\alpha}$$

Heavy tails are significant

Recent efforts have empirically tied the presence of strong heavy tails during training with good generalization performance.

- Simsekli, U., Sagun, L., Gürbüzbalaban, M. (2019). A Tail-Index Analysis of Stochastic Gradient Noise in Deep Neural Networks
- Martin, C. H., Peng, T., Mahoney, M. W. (2020). Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data.

Heavier tails imply wider exploration

A simple one-dimensional experiment

A 1D experiment

$$W_{k+1} = W_k - \gamma(A_k f'(W_k) + B_k)$$

A 1D experiment

$$W_{k+1} = W_k - \gamma(\underbrace{A_k}_{\text{multiplicative}} f'(W_k) + \underbrace{B_k}_{\text{additive}})$$

Compare

a. light additive noise ($B_k \sim \mathcal{N}(0, \sigma^2)$)

b. heavy additive noise ($B_k \sim \sigma t_\nu$)

c. multiplicative noise

$$(A_k \sim \mathcal{N}(1, \sigma^2), \quad B_k \sim \mathcal{N}(0, \epsilon^2))$$

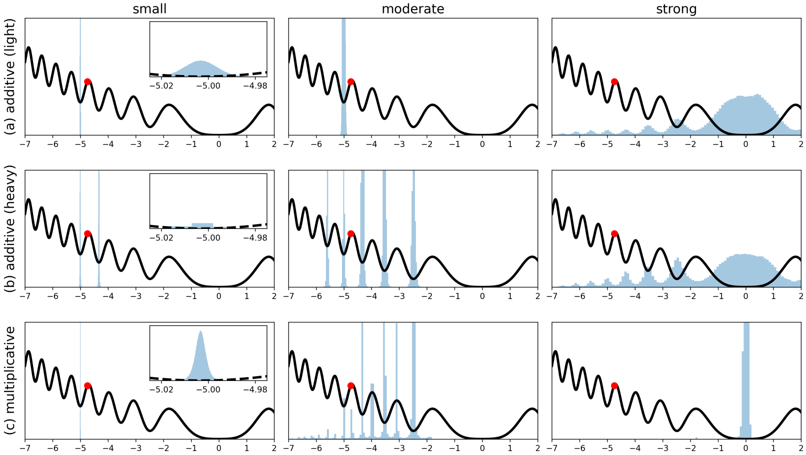


Figure: Histograms of 10^6 iterations of GD with combinations of small, moderate, and strong vs. light additive, heavy additive, and multiplicative noise, applied to a **non-convex objective** & initial starting location for the optimization.

Optimal search strategies

Optimizing the success of random searches

G. M. Viswanathan*†‡, Sergey V. Buldyrev*, Shlomo Havlin*§, M. G. E. da Luz‖¶, E. P. Raposo‖# & H. Eugene Stanley*

We address the general question of what is the best statistical strategy to adapt in order to search efficiently for randomly located objects ('target sites'). It is often assumed in foraging theory that the flight lengths of a forager have a characteristic scale: from this assumption gaussian, Rayleigh and other classical distributions with well-defined variances have arisen. However, such theories cannot explain the long-tailed power-law distributions[1,2] of flight lengths or flight times[3–6] that are observed experimentally. Here we study how the search efficiency depends on the probability distribution of flight lengths taken by a forager that can detect target sites only in its limited vicinity. We show that, when the target sites are sparse and can be visited any number of times, an inverse square power-law distribution of flight lengths, corresponding to Lévy flight motion, is an optimal strategy. We test the theory by analysing experimental foraging data on selected insect, mammal and bird species, and find that they are consistent with the predicted inverse square power-law distributions.

Lévy flights are characterized by a distribution function

$$P(l_j) \sim l_j^{-\mu} \quad (1)$$

with $1 < \mu \leq 3$, where l_j is the flight length. The gaussian is the stable distribution for the special case $\mu \geq 3$ owing to the central-limit theorem, while values $\mu \leq 1$ do not correspond to probability distributions that can be normalized[2]. This generalization, equation (1), introduces a natural parameter μ such that we essentially have a

"Since Levy flights and walks can optimize search efficiencies, therefore natural selection should have led to adaptations for Levy flight foraging"

Viswanathan, G.M., Raposo, E.P., da Luz, M.G.E. (2008). Levy flights and superdiffusion in the context of biological encounters and random searches. Physics of Life Reviews. 5(3): 133–150.

Viswanathan, G.M., Buldyrev, S.V., Havlin, S., Da Luz, M.G.E., Raposo, E.P. and Stanley, H.E., 1999. Optimizing the success of random searches. Nature, 401(6756), pp.911-914.

Establishing heavy tails

Ridge regression

Consider least squares linear regression with L^2 regularization:

$$M^* = \underset{M \in \mathbb{R}^{d \times m}}{\arg\min} \frac{1}{2}\mathbb{E}\|Y - MX\|^2 + \frac{1}{2}\lambda\|M\|_F^2,$$

where

- $X \in \mathbb{R}^d$ are the inputs
- $Y \in \mathbb{R}^m$ are the labels

Ridge regression

Lemma

The iterates M_k of **minibatch SGD** satisfy the following: for $W_k = \text{vec}(M_k)$,

$$W_{k+1} = A_k W_k + B_k,$$

where

$$A_k = I \otimes \left((1-\lambda) I - \gamma n^{-1} \sum_{i=1}^{n} X_{ik} X_{ik}^\top \right), \quad B_k = -\gamma n^{-1} \sum_{i=1}^{n} Y_{ik} X_{ik}^\top$$

There is both **additive** and **multiplicative** noise.

Kesten (1973): $\mathbb{P}(\sigma_{\min}(A_k) > 1) > 0 \implies$ heavy tails

Ridge regression

The ridge regression setting is covered in much greater detail in

Gurbuzbalaban, M., Simsekli, U., Zhu, L. (2020). The Heavy-Tail Phenomenon in SGD. arXiv:2006.04740.

The Kesten mechanism

Heavy tails (power laws) arise gradually **over time** due to the presence of noise on **multiple scales**

$$W_{k+1} = f_k(W_k) \approx A_k W_k + B_k$$

$\boldsymbol{A_k}$	$\boldsymbol{B_k}$
logarithmic scale	linear scale
multiplicative noise	additive noise
$D^1 f_k$	$D^0 f_k$

General stochastic optimization

In machine learning, solving problems of the form

$$w^* = \arg\min_w f(w), \quad f(w) := \mathbb{E}_{\mathcal{D}} \ell(w, X),$$

for a loss ℓ depending on weights w and data X from some dataset $\mathcal{D}$.

General stochastic optimization

Fixed point iteration: if Ψ is chosen such that fixed points of $\mathbb{E}_{\mathcal{D}}\Psi(\cdot, X)$ are minimizers of f, then

$$w_{k+1} = \mathbb{E}_{\mathcal{D}}\Psi(w_k, X)$$

either diverges, or converges to w^*.

General stochastic optimization

Estimating the expectation gives a **stochastic optimizer:**

$$W_{k+1} = \frac{1}{n} \sum_{i=1}^{n} \Psi(W_k, X_{ik}), \qquad X_{ik} \overset{\text{iid}}{\sim} X$$

where X_{ik} is the i-th datum from the k-th minibatch.

- Assuming data is shuffled in each epoch
- Forms a time-homogeneous Markov chain for fixed hyperparameters

General stochastic optimization

Estimating the expectation gives a **stochastic optimizer:**

$$\Psi(W_k, X_k)$$

$$W_{k+1} = \frac{1}{n}\sum_{i=1}^{n} \Psi(W_k, X_{ik}), \qquad X_k \overset{\text{iid}}{\sim} X.$$

- Assuming data is shuffled in each epoch
- Forms a time-homogeneous **Markov chain**

Stochastic optimization as a Markov chain

The sequence of **iterated random functions**

$$W_{k+1} = \Psi(W_k, X_k) \qquad X_k \overset{\text{iid}}{\sim} X.$$

Equivalently, as a root-finding problem:

$$W_{k+1} = W_k - \tilde{\Psi}(W_k, X_k) \qquad \text{(Borovkov)}$$

Assume this Markov chain is **ergodic**.

Diaconis, P., Freedman, D. (1999) Iterated Random Functions. SIAM Review. 41(1), 45–76.

Alsmeyer, G. (2003) On the Harris recurrence of iterated random Lipschitz functions and related convergence rate results. Journal of Theoretical Probability, 16(1):217–247,

Every iterative stochastic optimization algorithm in ML (with fixed hyperparameters) can be written as a Markov chain in this way.

SGD & SGD with momentum

Minibatch SGD: For minibatch size n and step size γ,

$$\Psi(w, X) = w - \gamma n^{-1} \sum_{i=1}^{n} \nabla \ell(w, X_i).$$

Momentum: Incorporating velocity v,

$$\Psi\left(\begin{pmatrix} v \\ w \end{pmatrix},\ X\right) = \frac{1}{n} \sum_{i=1}^{n} \begin{pmatrix} \eta v + \nabla \ell(w, X_i) \\ w - \gamma(\eta v + \nabla \ell(w, X_i)) \end{pmatrix}$$

Main Result

Theorem

Suppose X is non-atomic and there exist $k_\Psi, K_\Psi, M_\Psi, w^*$ such that as $\|w\| \to \infty$,

$$k_\Psi(X) - o(1) \leq \frac{\|\Psi(w, X) - \Psi(w^*, X)\|}{\|w - w^*\|} \leq K_\Psi(X) + o(1).$$

Main Result

Theorem

Suppose X is non-atomic and there exist k_Ψ, K_Ψ, M_Ψ, w^* such that as $\|w\| \to \infty$,

$$k_\Psi(X) - o(1) \le \frac{\|\Psi(w,X) - \Psi(w^*,X)\|}{\|w - w^*\|} \le K_\Psi(X) + o(1).$$

If $\mathbb{P}(k_\Psi(X) > 1) > 0$ and $\mathbb{E}\log K_\Psi(X) < 0$, for some $\mu, \nu, C_\mu, C_\nu > 0$,

$$C_\mu(1+t)^{-\mu} \le \mathbb{P}(\|W_\infty\| > t) \le C_\nu t^{-\nu}.$$

II. Factors influencing tail behaviour

Run SGD w/ constant step size on two-layer NN with L^2 loss using Wine Quality UCI dataset.

$\hat{\alpha}$ is an estimate of the tail exponent α such that

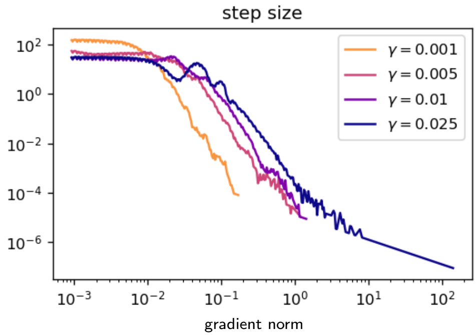
$$\mathbb{P}(\|D_\infty\| > t) \approx ct^{-\alpha}$$

- for fluctuations $D_k \doteq W_{k+1} - W_k$ (for SGD, corresponds to **gradient norm**)
- $D_\infty = \lim_{k\to\infty} D_k$ has the same tail exponent as W_k

Factors: step size

Prediction: larger step sizes $\Longrightarrow$ heavier tails

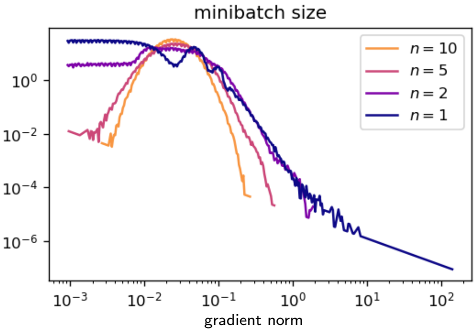
step size	
γ	$\hat{\alpha}$
0.001	4.12 ± 0.04
0.005	3.70 ± 0.02
0.01	3.71 ± 0.04
0.025	2.97 ± 0.03



Factors: minibatch size

Prediction: smaller batch sizes $\Longrightarrow$ heavier tails

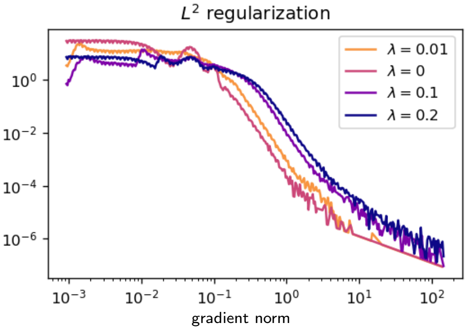
minibatch size	
n	$\hat{\alpha}$
10	5.99 ± 0.05
5	4.98 ± 0.07
2	3.62 ± 0.03
1	2.97 ± 0.03



Factors: L^2 regularization

Prediction: more regularization $\implies$ heavier tails

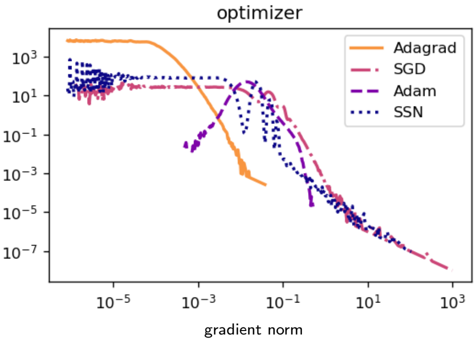
L^2 regularization	
λ	$\hat{\alpha}$
10^{-4}	2.97 ± 0.03
0.01	3.02 ± 0.02
0.1	2.77 ± 0.01
0.2	2.55 ± 0.01



Factors: optimizer

Prediction: SGD, SSN heavier than Adagrad, Adam

optimizer	
	$\hat{\alpha}$
Adagrad	3.2 ± 0.1
Adam	2.119 ± 0.005
SGD	2.93 ± 0.03
SSN	0.79 ± 0.04



Summary

Multiplicative noise is a critical element for understanding performance of stochastic optimizers

- Results in heavy-tailed stationary behaviour
- Far-reaching, but efficient, exploration

Future work:

- Improve precision for tail exponent estimates in more specific models (e.g. deep neural nets)
- The Kesten mechanism in the spectral domain
- Generalization bounds in discrete time

Hodgkinson, L., Mahoney, M. W. (2020) Multiplicative noise and heavy tails in stochastic optimization. arXiv:2006.06293

Outline

Heavy-tailed Self-regularization Theory

"Multiplicative noise and heavy tails in stochastic optimization," HM, ICML 2021

"Generalization Properties of Stochastic Optimizers via Trajectory Analysis," HSKM, ICML 2022

When are ensembles really effective?

What are generalization bounds?

Empirical Risk Minimization

To train parameterized models, solve

$$w^* = \arg\min_w \mathcal{R}_n(w), \;\; \mathcal{R}_n(w) := \frac{1}{n} \sum_{i=1}^{n} \ell(w, X_i),$$

for a loss ℓ depending on weights w and data $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{D}$.

Generalization bounds

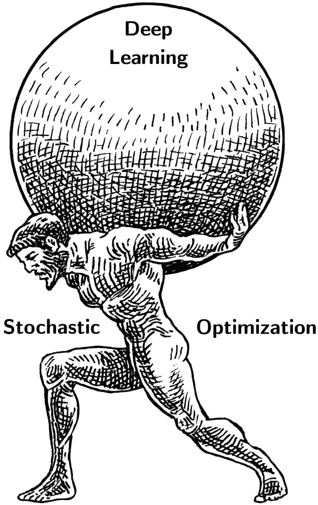
Bounds on the excess risk

$$\mathcal{E}_n(w^*) = \mathcal{R}_n(w^*) - \underbrace{\mathbb{E}_{\mathcal{D}}\mathcal{R}_n(w^*)}_{\text{generalization}}$$

Stochastic optimization

is the process of minimizing an objective function via the simulation of random elements.

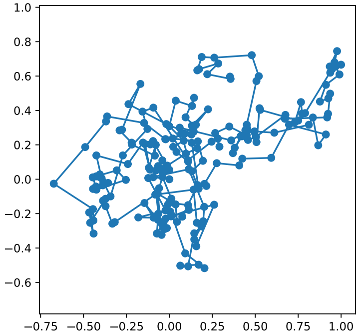
"the backbone of modern machine learning"



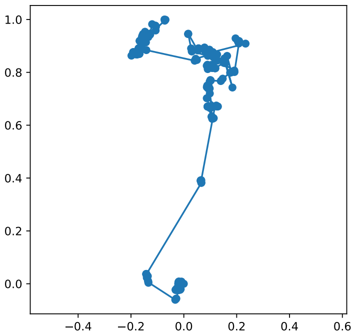
How do the dynamics of the optimizer influence generalization?

Types of Dynamics

Brownian motion
light-tailed

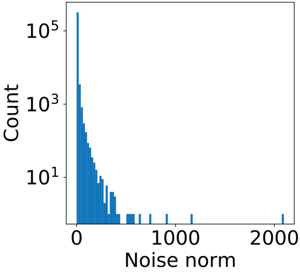


Lévy flight
heavy-tailed

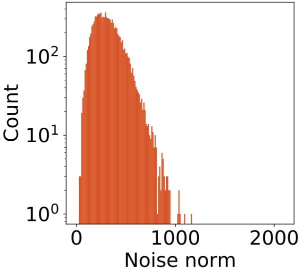


Heavy Tails in Machine Learning

Norms of optimizer steps in a deep learning task



(a) Real



(b) Gaussian

Şimşekli, U., Sagun, L., & Gurbuzbalaban, M. (2019, May). A tail-index analysis of stochastic gradient noise in deep neural networks. In International Conference on Machine Learning (pp. 5827-5837). PMLR.

Previous Work

Under a **(continuous-time) Feller process model** of SGD,

$$\text{heavier tails} \implies \text{smaller } \mathcal{E}_n.$$

Şimşekli, U., Sener, O., Deligiannidis, G., & Erdogdu, M. A. (2020). Hausdorff dimension, heavy tails, and generalization in neural networks. Advances in Neural Information Processing Systems, 33, 5138-5151.

- Complicated assumptions
- What about **discrete time**, i.e. SGD itself?

Markov Assumption

Assume that the iterates of the optimizer

$$W_1, W_2, \ldots, W_k, \ldots$$

are a **Markov chain**.

Upper Tail Exponent

Previous works have considered the **upper tail exponent**:

$$\mathbb{P}(\|W_{k+1} - W_k\| > r) \approx \mathcal{O}(r^{-\beta}).$$

as $r \to \infty$.

Lower Tail Exponent

What about the **lower tail exponent**?

$$\mathbb{P}(\|W_{k+1} - W_k\| \leq r) \approx \mathcal{O}(r^{\alpha}).$$

as $\boldsymbol{r} \to 0^{+}$.

Lower Tail Exponent

Theorem (Informal)

Assume that iterates W_k of an optimizer satisfy

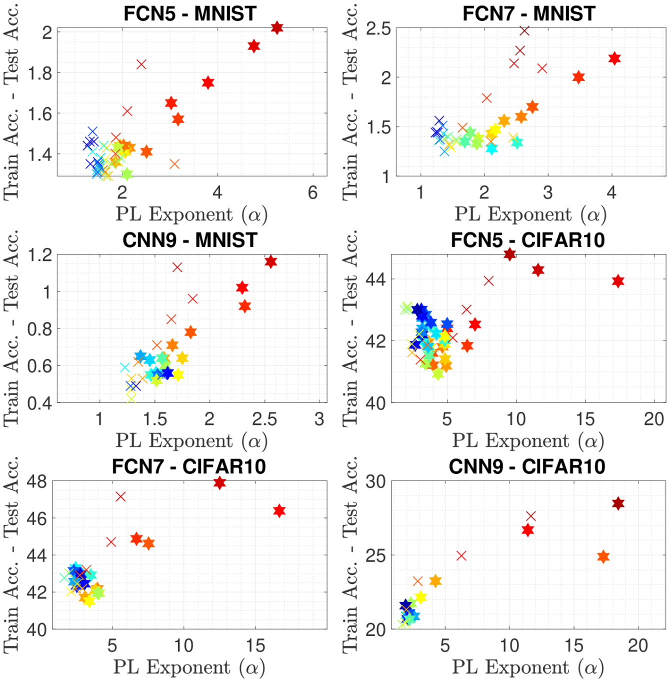
$$\mathbb{P}(\|W_{k+1} - W_k\| \leq r) \approx \mathcal{O}(r^\alpha)$$

in the neighbourhood of a local optimum w^*. Then an upper bound on

$$\mathbb{E} \sup_{k=1,\ldots,m} |\mathcal{E}_n(W_k)| \text{ is positively correlated with } \alpha.$$

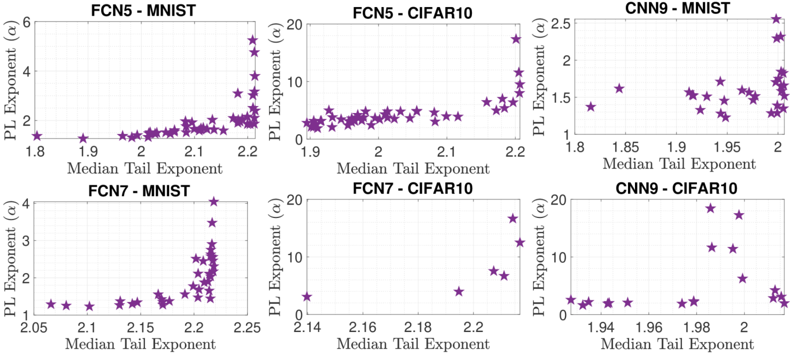
Is this true in practice?

Train NNs with varying hyperparameters & regularization


FCN5 - MNIST
FCN7 - MNIST
CNN9 - MNIST
FCN5 - CIFAR10
FCN7 - CIFAR10
CNN9 - CIFAR10
Train Acc. - Test Acc.
PL Exponent (α)

Lower Tail Exponent

Lower tail often correlates with upper tail



Summary

- Developed a **general proof technique** for linking optimizer dynamics to generalization
- Extended results of Șimșekli et al., 2020.
- Lower tail exponent correlates with $\mathcal{E}_n$
  - Supported in practice
  - Lower tail correlates with upper tail

Outline

Heavy-tailed Self-regularization Theory

"Multiplicative noise and heavy tails in stochastic optimization," HM, ICML 2021

"Generalization Properties of Stochastic Optimizers via Trajectory Analysis," HSKM, ICML 2022

When are ensembles really effective?

Taxonomizing loss landscapes

Taxonomizing local versus global structure in neural network loss landscapes, Yang et al. arXiv:2107.11228

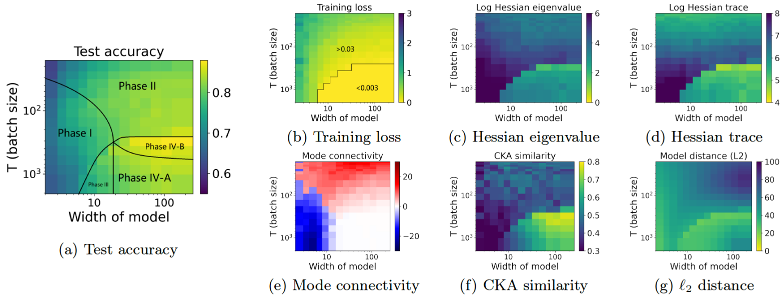


Figure 2: **(Standard setting).** Partitioning the 2D load-like—temperature-like diagram into different phases of learning, using batch size as the temperature and varying model width to change load. Models are trained with ResNet18 on CIFAR-10. All plots are on the same set of axes. We note that batch size is inverse temperature, and thus it has smaller values at the top of the y-axis and larger values at the bottom.

Taxonomizing loss landscapes

Taxonomizing local versus global structure in neural network loss landscapes, Yang et al. arXiv:2107.11228

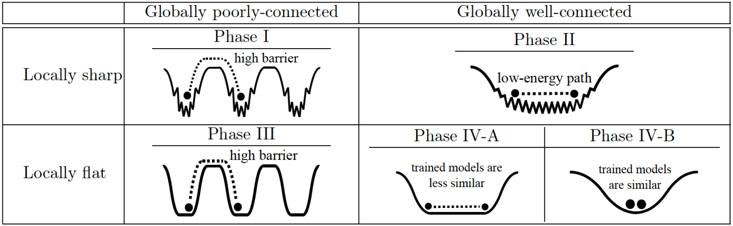
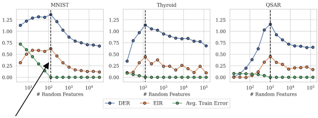


Figure 1: **(Caricature of different types of loss landscapes).** Globally well-connected versus globally poorly-connected loss landscapes; and locally sharp versus locally flat loss landscapes. Globally well-connected loss landscapes can be interpreted in terms of a global "rugged convexity"; and globally well-connected and locally flat loss landscapes can be further divided into two sub-cases, based on the similarity of trained models.

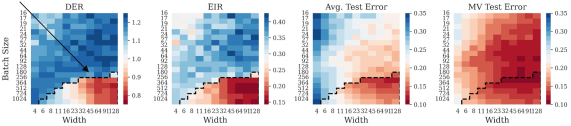
The weakness of modern weak learners?

Ensembling?

Bagged Random
Feature classifiers



Interpolation threshold



ResNet18/CIFAR-10
Deep Ensembles

"When are ensembles really effective?," Theisen, et al. arXiv:2305.12313 (2023)

Theory:

- Characterize the "ensemble improvement rate" in terms of the "disagreement-error ratio"
- If disagreement > average error, then ensembles improve performance when DER is large
- If disagreement < average error, then ensembles do not improve performance too much when DER is small

Empirical:

- Ensemble improvement, DER become small beyond the "interpolation" threshold
- Ensembling becomes less useful for large models which can easily "interpolate" the training
- This corresponds to the disagreement-error ratio getting smaller in this regime

Contributions and Conclusions

- For modern ML models, weights are HT, gradients are HT, etc are HT
- HTs are hard
- Can *use* this theory to:
  - predict trends in the quality of SOTA neural networks without access to training or testing data
  - perform diagnostics at scale, including identifing Simpson's paradoxes in public benchmarks
  - predict overfitting/underfitting
  - characterize benefits/non-benefits of ensembling
- Seems worth considering more ...