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.
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.

Phases of Training

**Exploration**
large learning rate

**Exploitation**
small learning rate

A distributional approach

Investigate how a stochastic optimizer explores the loss landscape

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Model stochastic optimization as a random dynamical system (Markov)</td>
</tr>
<tr>
<td>2.</td>
<td>Fix all hyperparameters to particular values (time-homogeneous; no annealing)</td>
</tr>
<tr>
<td>3.</td>
<td>Examine properties of the <strong>stationary</strong> (invariant) distribution</td>
</tr>
</tbody>
</table>

▶ 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.

\[ \mathbb{P}(W > t) \approx ct^{-\alpha} \]
Recent efforts have empirically tied the presence of strong heavy tails during training with good generalization performance.


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

\[ W_{k+1} = W_k - \gamma \left( A_k f'(W_k) + B_k \right) \]

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.
Optimizing the success of random searches


We address the general question of what is the best statistical strategy to adopt 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

\[ P(l) \sim l^{-\mu} \]

with \(1 < \mu \leq 3\), where \(l\) 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\(^5\). This generalization, equation (1), introduces a natural parameter \(\mu\) 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”


Establishing heavy tails
Consider least squares linear regression with $L^2$ regularization:

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

where

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

The Kesten mechanism

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

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

<table>
<thead>
<tr>
<th>$A_k$</th>
<th>$B_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>logarithmic scale</td>
<td>linear scale</td>
</tr>
<tr>
<td>multiplicative noise</td>
<td>additive noise</td>
</tr>
<tr>
<td>$D^1 f_k$</td>
<td>$D^0 f_k$</td>
</tr>
</tbody>
</table>
In machine learning, solving problems of the form

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

for a loss \( \ell \) depending on weights \( w \) and data \( X \) from some dataset \( D \).
Fixed point iteration: if $\Psi$ is chosen such that fixed points of $\mathbb{E}_D \Psi(\cdot, X)$ are minimizers of $f$, then

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

either diverges, or converges to $w^*$. 

General stochastic optimization

Estimating the expectation gives a stochastic optimizer:

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

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:

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

- Assuming data is shuffled in each epoch
- Forms a time-homogeneous Markov chain
The sequence of iterated random functions

\[ W_{k+1} = \Psi(W_k, X_k) \quad X_k \text{iid} \sim X. \]

Equivalently, as a root-finding problem:

\[ W_{k+1} = W_k - \tilde{\Psi}(W_k, X_k) \quad \text{(Borovkov)} \]

Assume this Markov chain is ergodic.


Every iterative stochastic optimization algorithm in ML (with fixed hyperparameters) can be written as a Markov chain in this way.
**Minibatch SGD:** For minibatch size $n$ and step size $\gamma$, 

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

**Momentum:** Incorporating velocity $v$, 

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

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

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

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

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

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

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

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

- for fluctuations $D_k = W_{k+1} - W_k$ (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 $\Rightarrow$ heavier tails

<table>
<thead>
<tr>
<th>step size</th>
<th>$\hat{\alpha}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma = 0.001$</td>
<td>$4.12 \pm 0.04$</td>
</tr>
<tr>
<td>$\gamma = 0.005$</td>
<td>$3.70 \pm 0.02$</td>
</tr>
<tr>
<td>$\gamma = 0.01$</td>
<td>$3.71 \pm 0.04$</td>
</tr>
<tr>
<td>$\gamma = 0.025$</td>
<td>$2.97 \pm 0.03$</td>
</tr>
</tbody>
</table>
Factors: minibatch size

Prediction: smaller batch sizes $\implies$ heavier tails

<table>
<thead>
<tr>
<th>minibatch size</th>
<th>$\hat{\alpha}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>$5.99 \pm 0.05$</td>
</tr>
<tr>
<td>5</td>
<td>$4.98 \pm 0.07$</td>
</tr>
<tr>
<td>2</td>
<td>$3.62 \pm 0.03$</td>
</tr>
<tr>
<td>1</td>
<td>$2.97 \pm 0.03$</td>
</tr>
</tbody>
</table>

minibatch size

gradient norm

$n = 10$

$n = 5$

$n = 2$

$n = 1$
Factors: \( L^2 \) regularization

**Prediction:** more regularization \( \Rightarrow \) heavier tails

<table>
<thead>
<tr>
<th>( L^2 ) regularization</th>
<th>( \lambda )</th>
<th>( \hat{\alpha} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10(^{-4})</td>
<td>2.97 ± 0.03 (\hat{\alpha})</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>3.02 ± 0.02 (\hat{\alpha})</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>2.77 ± 0.01 (\hat{\alpha})</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>2.55 ± 0.01 (\hat{\alpha})</td>
</tr>
</tbody>
</table>

\( L^2 \) regularization

![Graph showing the effect of different \( \lambda \) values on \( \hat{\alpha} \) for gradient norm](image-url)
Factors: optimizer

Prediction: SGD, SSN heavier than Adagrad, Adam

<table>
<thead>
<tr>
<th>optimizer</th>
<th>$\hat{\alpha}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adagrad</td>
<td>3.2 ± 0.1</td>
</tr>
<tr>
<td>Adam</td>
<td>2.119 ± 0.005</td>
</tr>
<tr>
<td>SGD</td>
<td>2.93 ± 0.03</td>
</tr>
<tr>
<td>SSN</td>
<td>0.79 ± 0.04</td>
</tr>
</tbody>
</table>
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

Outline

Heavy-tailed Self-regularization Theory

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

“When are ensembles really effective?”

“Generalization Properties of Stochastic Optimizers via Trajectory Analysis,” HSKM, ICML 2022
What are generalization bounds?
To train parameterized models, solve

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

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

Bounds on the excess risk

\[ \mathcal{E}_n(w^*) = \mathcal{R}_n(w^*) - \mathbb{E}_D \mathcal{R}_n(w^*) \]

\( w^* \) 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

Under a \textit{(continuous-time) Feller process model} of SGD,

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


- Complicated assumptions
- What about \textbf{discrete time}, i.e. SGD itself?
Assume that the iterates of the optimizer

\[ W_1, W_2, \ldots, W_k, \ldots \]

are a Markov chain.
Previous works have considered the upper tail exponent:

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

as \( r \to \infty \).
What about the lower tail exponent?

\[ P\left( \| W_{k+1} - W_k \| \leq r \right) \approx O(r^\alpha). \]

as \( r \to 0^+ \).
Theorem (Informal)

Assume that iterates $W_k$ of an optimizer satisfy

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

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

$$
\mathbb{E} \sup_{k=1,\ldots,m} |\mathcal{E}_n(W_k)|
$$

is positively correlated with $\alpha$. 

Is this true in practice?

*Train NNs with varying hyperparameters & regularization*
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 Şimşekli 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?
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*

<table>
<thead>
<tr>
<th></th>
<th>Globally poorly-connected</th>
<th>Globally well-connected</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Locally sharp</strong></td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
<tr>
<td>Phase I</td>
<td>high barrier</td>
<td>low-energy path</td>
</tr>
<tr>
<td>Phase III</td>
<td>high barrier</td>
<td>trained models are less similar</td>
</tr>
<tr>
<td>Phase IV-A</td>
<td></td>
<td>trained models are similar</td>
</tr>
<tr>
<td>Phase IV-B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

**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

"When are ensembles really effective?,” Theisen, et al. arXiv:2305.12313 (2023)
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 identifying Simpson's paradoxes in public benchmarks
  - predict overfitting/underfitting
  - characterize benefits/non-benefits of ensembling
- Seems worth considering more ...