Journal of Machine Learning Research 25 (2024) 1-20 Submitted 10/23; Revised 4/24; Published 6/24

Sharpness-Aware Minimization and the Edge of Stability

Philip M. Long PLONG@GOOGLE.COM
Peter L. Bartlett* PETERBARTLETT@GOOGLE.COM
Google
1600 Amphitheatre Parkway
Mountain View, CA 94040

Editor: Francis Bach

Abstract

Recent experiments have shown that, often, when training a neural network with gradient descent (GD) with a step size η, the operator norm of the Hessian of the loss grows until it approximately reaches $2/\eta$, after which it fluctuates around this value.

The quantity $2/\eta$ has been called the "edge of stability" based on consideration of a local quadratic approximation of the loss. We perform a similar calculation to arrive at an "edge of stability" for Sharpness-Aware Minimization (SAM), a variant of GD which has been shown to improve its generalization. Unlike the case for GD, the resulting SAM-edge depends on the norm of the gradient. Using three deep learning training tasks, we see empirically that SAM operates on the edge of stability identified by this analysis.

Keywords: Sharpness-aware minimization, edge of stability, optimization, deep learning, wide minima.

1. Introduction

Sharpness-aware Minimization (SAM) (Foret et al., 2020) is a new gradient-based neural network training algorithm that advanced the state-of-the-art test accuracy on a number of prominent benchmark datasets. As its name suggests, it explicitly seeks to find a solution that not only fits the training data, but that avoids "sharp" minima, for which nearby parameter vectors perform poorly. SAM is an incremental algorithm that updates its parameters using a gradient computed at a neighbor of the current solution. The neighbor is the point in parameter space found by taking a step of length ρ "uphill" in the gradient direction. The practical success of SAM has motivated theoretical research (Bartlett et al., 2023; Wen et al., 2023; Andriushchenko et al., 2023), including results highlighting senses in which SAM's update may be viewed, under certain conditions, as including a component that performs gradient descent on the operator norm of the Hessian (Bartlett et al., 2023; Wen et al., 2023).

Meanwhile, Cohen et al. (2021), building on the work of Jastrzebski et al. (2019) and others, exposed a striking phenomenon regarding neural network training with the original gradient descent (GD) method: for many initialization schemes and learning rates η, the

*. Also affiliated with University of California, Berkeley.

operator norm of the Hessian eventually settles in the neighborhood of $2/\eta$. This has been termed the "edge of stability", in part because a convex quadratic trained by gradient descent with a learning rate η will only converge if the operator norm of its Hessian (which is the same everywhere) is less than $2/\eta$. This phenomenon also inspired substantial theoretical research (see Arora et al., 2022; Damian et al., 2023; Ma et al., 2022; Zhu et al., 2022; Ahn et al., 2022; Chen and Bruna, 2022). One result identified conditions under which, when training approaches the edge of stability, the dynamics includes a self-stabilization mechanism that tends to drive the operator norm of the Hessian back down (Damian et al., 2023).

In this paper, we investigate whether SAM operates at the edge of stability. First, we perform a derivation, analogous to the one that identifies $2/\eta$ as the edge of stability for GD, that yields a formula for the operator norm of the Hessian that may be viewed as the edge of stability for SAM. As expected, SAM's edge of stability depends on the radius ρ of its neighborhood. It also depends on the norm of the gradient of the training error at the current solution, unlike the case of GD. As the norm of the gradient gets smaller, the edge of stability for SAM also gets smaller.

Next, we evaluate experimentally whether SAM operates at the edge of stability identified by our analysis. Our first experiments are with fully connected networks on MNIST. Here, it is feasible to experiment with a version of SAM that uses a batch gradient, albeit computed at the neighbor uphill of the current iterate at a distance ρ. For many combinations of the step size η and the radius ρ, the operator norm of the Hessian at SAM's iterates closely matches the value arising from our analysis. Next, we experiment with a convolutional neural network training on 1000 examples from CIFAR10. Here again, we see SAM operating on the edge of stability. Finally, we experiment with a standard Transformer architecture training a language model on `tiny_shakespeare` using the more practical version of SAM that uses stochastic gradients. Here, we also see substantial agreement with our theoretical analysis.

In our experiments with SAM, its edge of stability is often *much* smaller than $2/\eta$, even early in training. Rather than first driving the training error to a very small value, and then drifting along a manifold of near-optimal solutions to wider minima, SAM's process drives solutions toward smooth regions of parameter space early in training, while the loss is still large.

The derivation of SAM's edge of stability is in Section 2. The experiments are described in detail in Section 3. The results are in Section 4. Section 5 includes further description of related work. We conclude in Section 6.

2. Derivation

The *Sharpness-Aware Minimization* algorithm is defined by the update

$$w_{t+1} = w_t - \eta \nabla \ell \left(w_t + \rho \frac{\nabla \ell(w_t)}{\|\nabla \ell(w_t)\|} \right). \tag{1}$$

This is like gradient descent, except using a gradient evaluated at $w_t + \rho \frac{\nabla \ell(w_t)}{\|\nabla \ell(w_t)\|}$ instead of w_t.

In this section, we calculate an "edge of stability" for SAM analogous to the $2/\eta$ value for GD.

Before analyzing SAM, however, let us review the standard analysis that identifies the edge of stability for GD, assuming for simplicity that the quadratic approximation around an iterate is exact.

Proposition 1 *For $w_t \in \mathbb{R}^d$, $\eta > 0$, if*

- *$g = \nabla \ell(w_t) \neq 0$, $H = \nabla^2 \ell(w_t)$, $w_{t+1} = w_t - \eta g$, and*
- *for all $w \in \mathbb{R}^d$, $\ell(w) = \ell(w_t) + g^T(w - w_t) + \frac{(w-w_t)^\top H (w-w_t)}{2}$,*

then

- *if $||H||_{op} < \frac{2}{\eta}$, then $\ell(w_{t+1}) < \ell(w_t)$, and*
- *this condition on $||H||_{op}$ is the weakest possible of its type: if*
  - *g is aligned with a principal eigenvector of H whose eigenvalue is non-negative, then*
  - *$\text{sign}(\ell(w_{t+1}) - \ell(w_t)) = \text{sign}\left(||H||_{op} - \frac{2}{\eta}\right)$.*

Proof Substituting $w_{t+1} - w_t$ into the formula for ℓ, we have

$$\begin{aligned}
\ell(w_{t+1}) &= \ell(w_t) - \eta g^\top g + \frac{\eta^2 g^\top H g}{2} \\
&\leq \ell(w_t) - \eta g^\top g + \frac{\eta^2 g^\top ||H||_{op} g}{2} \\
&= \ell(w_t) - \eta \left(1 - \frac{\eta ||H||_{op}}{2}\right) ||g||^2.
\end{aligned}$$

If $||H||_{op} < \frac{2}{\eta}$, since $g \neq 0$, this implies $\ell(w_{t+1}) < \ell(w_t)$.

When g is aligned with a principal eigenvector of H whose eigenvalue is non-negative, we have $Hg = ||H||_{op} g$, which implies, as above, that

$$\ell(w_{t+1}) = \ell(w_t) - \eta \left(1 - \frac{\eta ||H||_{op}}{2}\right) ||g||^2,$$

which, again since $g \neq 0$, implies $\text{sign}(\ell(w_{t+1}) - \ell(w_t)) = \text{sign}\left(||H||_{op} - \frac{2}{\eta}\right)$. ■

We can think of Proposition 1 as formalizing the statement that $2/\eta$ is the edge of stability for GD: if $||H||_{op} < 2/\eta$, GD is guaranteed to make progress, and no larger bound suffices.

Even in the convex quadratic case, the dynamics of SAM are much more complex than GD (see Bartlett et al., 2023). However, if we bound $||H||_{op}$ in terms of $||g||$ as well as η and ρ, an analogous result holds.

Proposition 2 *For $w_t \in \mathbb{R}^d$, $\eta > 0$, $\rho > 0$, if*

- *$g = \nabla \ell(w_t) \neq 0$, $H = \nabla^2 \ell(w_t) \succeq 0$, $w_{t+1} = w_t - \eta \nabla \ell \left(w_t + \rho \frac{\nabla \ell(w_t)}{\|\nabla \ell(w_t)\|}\right)$, and*

- *for all* $w \in \mathbb{R}^d$, $\ell(w) = \ell(w_t) + g^T(w - w_t) + \frac{(w-w_t)^\top H(w-w_t)}{2}$,

then

- *if* $||H||_{op} < \frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right)$, *then* $\ell(w_{t+1}) < \ell(w_t)$, *and*
- *this condition on* $||H||_{op}$ *is the weakest possible of its type: if*
  - g *is aligned with a principal eigenvector of* H, *then*
  - $\text{sign}(\ell(w_{t+1}) - \ell(w_t)) = \text{sign}\left(||H||_{op} - \frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right)\right)$.

Proposition 2 is an immediate consequence of the following stronger, but somewhat more technical, proposition.

Proposition 3 *For* $w_t \in \mathbb{R}^d$, $\eta > 0$, $\rho > 0$, *if*

- $g = \nabla\ell(w_t) \neq 0$ *and* $H = \nabla^2\ell(w_t)$ *has eigenvalues* $\lambda_1, ..., \lambda_d$ *and unit-length eigenvectors* $v_1, ..., v_d$,
- $w_{t+1} = w_t - \eta\nabla\ell\left(w_t + \rho\frac{\nabla\ell(w_t)}{\|\nabla\ell(w_t)\|}\right)$,
- *for all* $w \in \mathbb{R}^d$, $\ell(w) = \ell(w_t) + g^T(w - w_t) + \frac{(w-w_t)^\top H(w-w_t)}{2}$

then

- *if, for all* i,
$$-\frac{||g||}{\rho} \leq \lambda_i \leq \frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right),$$
and there is an i *such that*
$$g \cdot v_i \neq 0 \text{ and } -\frac{||g||}{\rho} < \lambda_i < \frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right),$$
then $\ell(w_{t+1}) < \ell(w_t)$, *and*
- *if*
  - g *is aligned with a principal eigenvector of* H *whose eigenvalue is non-negative, then*
  - $\text{sign}(\ell(w_{t+1}) - \ell(w_t)) = \text{sign}\left(||H||_{op} - \frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right)\right)$.

Proof Substituting $w_{t+1} - w_t$ into the formula for ℓ, in part since H is symmetric, we have

$$\begin{aligned}\ell(w_{t+1}) &= \ell(w_t) - \eta g^\top\left(g + \rho H\frac{g}{||g||}\right) + \frac{\eta^2\left(g + \rho H\frac{g}{||g||}\right)^\top H\left(g + \rho H\frac{g}{||g||}\right)}{2} \\ &= \ell(w_t) - \eta g^\top\left(I + \frac{\rho H}{||g||} - \eta\left(\frac{\left(I + \frac{\rho H}{||g||}\right)^2 H}{2}\right)\right)g.\end{aligned}$$

Using the fact that, since H is symmetric, any matrix polynomial of H has the same eigenvectors as H, we have

$$\begin{aligned}\ell(w_{t+1}) &= \ell(w_t) - \eta \sum_{i=1}^{n} (v_i \cdot g)^2 \left(1 + \frac{\rho\lambda_i}{||g||} - \eta \left(\frac{\left(1 + \frac{\rho\lambda_i}{||g||}\right)^2 \lambda_i}{2}\right)\right) \\ &= \ell(w_t) - \eta \sum_{i=1}^{n} (v_i \cdot g)^2 \left(1 + \frac{\rho\lambda_i}{||g||}\right)\left(1 - \frac{\eta\left(1 + \frac{\rho\lambda_i}{||g||}\right)\lambda_i}{2}\right). \end{aligned} \tag{2}$$

Recalling that each $\lambda_i \geq -\frac{||g||}{\rho}$, let us focus on the last factor of one term in the sum of (2) for which $\lambda_i > -\frac{||g||}{\rho}$ and $(v_i \cdot g)^2 \neq 0$. We have

$$\begin{aligned} &1 - \frac{\eta\left(1 + \frac{\rho\lambda_i}{||g||}\right)\lambda_i}{2} \geq 0 \\ &\Leftrightarrow \eta\rho\lambda_i^2 + \eta\lambda_i||g|| - 2||g|| \leq 0. \end{aligned}$$

The convex quadratic on the LHS has two solutions, one that is negative, and one that is positive:

$$\begin{aligned} &\frac{\pm\sqrt{\eta^2||g||^2 + 8\eta\rho||g||} - \eta||g||}{2\eta\rho} \\ &= \frac{||g||}{2\rho}\left(\pm\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right). \end{aligned}$$

Thus, given that $\lambda_i > -\frac{||g||}{\rho}$, the ith term of the sum in (2) is positive iff

$$-\frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} + 1\right) < \lambda_i < \frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right), \tag{3}$$

for which

$$-\frac{||g||}{\rho} < \lambda_i < \frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right),$$

suffices. Thus each term in the sum of (2) is non-negative, and at least one is positive, so $\ell(w_{t+1}) < \ell(w_t)$.

If g is aligned with a principal eigenvector of H whose eigenvalue is non-negative, assuming wlog that this principal eigenvector is v_1, we have $(v_1 \cdot g)^2 > 0$, and $(v_i \cdot g)^2 = 0$ for

all $i \neq 1$. In this case, all of the terms in the sum in (2) are zero except the first, thus

$$\begin{aligned}\text{sign}(\ell(w_{t+1}) - \ell(w_t)) &= -\text{sign}\left(1 - \frac{\eta\left(1 + \frac{\rho\lambda_1}{||g||}\right)\lambda_1}{2}\right) \\ &= \text{sign}\left(\lambda_1 - \frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right)\right) \\ &= \text{sign}\left(||H||_{op} - \frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right)\right),\end{aligned}$$

where we have used the equivalent bounds on λ_1 given by (3). ■

We refer to the threshold $\frac{||g||}{2\rho}\left(\sqrt{1 + \frac{8\rho}{\eta||g||}} - 1\right)$ identified in Proposition 2 as *SAM's edge of stability*, or the SAM-edge for short.

The ratio $\frac{||H||_{op}}{2/\eta}$ between the edge of stability for SAM, and the edge for GD, is

$$\frac{||H||_{op}}{2/\eta} = \frac{\eta\|g\|}{4\rho}\left(\sqrt{1 + \frac{8\rho}{\eta\|g\|}} - 1\right).$$

This ratio depends on η, ρ and $||g||$ through $\eta\|g\|/(2\rho)$; let us refer to this intermediate quantity as α. Figure 1 shows the function

$$\alpha \mapsto \frac{\alpha}{2}\left(\sqrt{1 + \frac{4}{\alpha}} - 1\right),$$

that, at SAM's edge of stability, gives $\|H\|_{op}/(2/\eta)$ as a function of $\alpha = \eta\|g\|/(2\rho)$. Notice that as $\alpha \to \infty$, this function approaches 1, and it approaches zero like $\sqrt{\alpha}$.

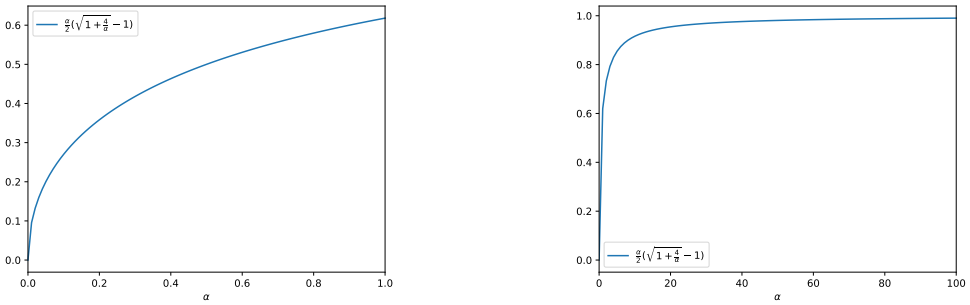


Figure 1: The ratio of SAM's edge of stability to $2/\eta$, the edge of stability for GD, as a function of $\alpha = \eta\|g\|/(2\rho)$.

Proposition 3 focuses on the case where the largest eigenvalue is positive. This is motivated in part by the work of Ghorbani et al. (2019), who found that, often, after a small

amount of training of a neural network, any negative eigenvalues in the Hessian are very small.

3. Methods

We performed experiments in three settings. In each setting, we trained for a variety of combinations of hyperparameters, and tracked various quantities, including the operator norm of the Hessian, and the SAM edge. Code is available (Long and Bartlett, 2024).

3.1 Settings

First, we trained a depth-four fully connected network, with 1000 nodes in each hidden layer, on MNIST using the quadratic loss with batch gradient descent. We trained for eight hours of wallclock time on a V100 GPU. The weights were initialized using Glorot normal initialization. Prior to training, the data was centered.

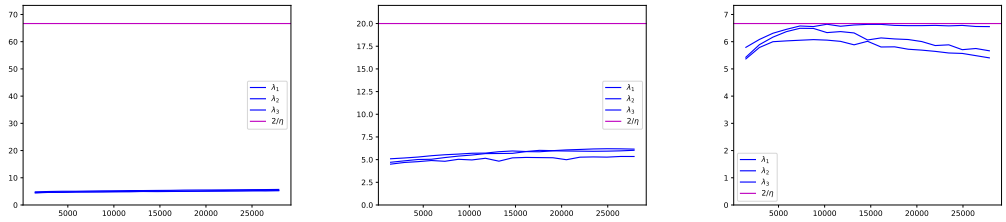
Next, we trained a CNN on CIFAR10 using the quadratic loss. To make batch gradients feasible, we only trained on the first 1000 examples. The CNN architecture was standard: there were two blocks comprised of a convolutional layer with a ReLU nonlinearity followed by layer normalization, then 2×2 max pooling with a 2×2 stride. In the first block the convolutional layer had 16 channels, and in the second block, it had 32 channels. Training was performed for 12 hours on a V100 GPU. Here again, the weights were initialized using Glorot normal initialization, and data was centered before training.

For the final setting, we modified the sample implementation of Transformers distributed with the Haiku package (see Hennigan et al., 2023), training an autoregressive character language model using the `tiny_shakespeare` dataset, using minibatches of size 128. The operator norm of the Hessian, and its principal directions, were also estimated using minibatches. The architecture was as in the Haiku distribution, with 6 layers, 8 heads, a key size of 32, "model size" of 128, and sequence length of 64. Because it introduces noise, Dropout was removed. The last 10000 lines of `tiny_shakespeare` were set aside as a test set, and the remaining data was used for training.

3.2 Hyperparameters

We trained once for each combination of the following hyperparameters:

- For MNIST,
  - learning rates η: 0.03, 0.1, 0.3,
  - SAM offsets ρ (see (1)): 0.0, 0.1, 0.3, 1.0.
- For CIFAR10,
  - learning rates: 0.0003, 0.001, 0.003, 0.01,
  - ρ values: 0.0, 0.1, 0.3, 1.0
- For `tiny_shakespeare`,
  - learning rates: 0.01, 0.02, 0.05, 0.1, 0.2, 0.5



(a) $\eta = 0.03$ (b) $\eta = 0.1$ (c) $\eta = 0.3$

Figure 2: Magnitudes of the largest eigenvalues of the Hessian when an MLP is trained with GD on MNIST.

– ρ values: 0.0, 0.1, 0.3, 1.0.

Results were discarded whenever training diverged.

3.3 Implementation

We coded our experiments using Jax (Bradbury et al., 2018), along with Flax (Heek et al., 2023) (for the image classification experiments), and Haiku (Hennigan et al., 2020) (for the language model experiments).

3.4 Unreported preliminary experiments

During an exploration phase, we conducted a number of preliminary experiments, during which we identified new statistics to collect, what hyperparameter combinations to try, etc. (For example, we wanted to minimize the fraction of runs with learning rates too small to bring about the edge of stability, and those with learning rates so large that training diverged.) The results reported in this paper were one series of final runs for the last combinations of hyperparameters.

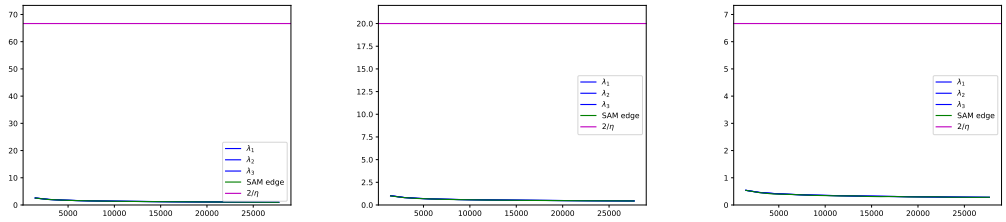
4. Results

All of the results from every run that did not diverge may be found in a supplementary folder. (In all of the plots, the training time in seconds is plotted along the horizontal axis.) In this section, we go over some of the most noteworthy results.

4.1 MNIST

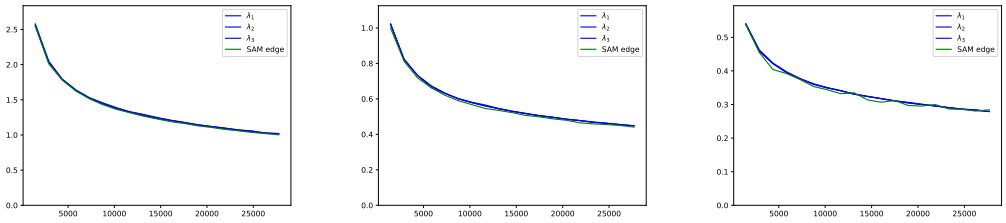
Figure 2 contains plots of the magnitudes of the top three eigenvalues of the Hessian, along with $2/\eta$ and the SAM-edge, when an MLP was trained on MNIST using gradient descent. There is a plot for each learning rate η. As reported by Cohen et al. (2021), if the learning rate is large enough, the operator norm of the Hessian stabilizes near $2/\eta$. We can think of GD as a special case of SAM with $\rho = 0$; the SAM-edge is of course $2/\eta$ in that case.

Figure 3 contains the analogous plots when $\rho = 0.1$. Despite the fact that gradients are



(a) $\eta = 0.03$ (b) $\eta = 0.1$ (c) $\eta = 0.3$

Figure 3: Magnitudes of the largest eigenvalues of the Hessian when an MLP is trained with SAM on MNIST, with $\rho = 0.1$.



(a) $\eta = 0.03$ (b) $\eta = 0.1$ (c) $\eta = 0.3$

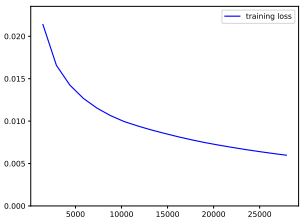
Figure 4: Magnitudes of the largest eigenvalues of the Hessian when an MLP is trained with SAM on MNIST, with $\rho = 0.1$.

taken from locations at a distance just 0.1 from each of the iterates, the cumulative effect results in solutions with Hessians an order of magnitude smaller than those seen with GD.

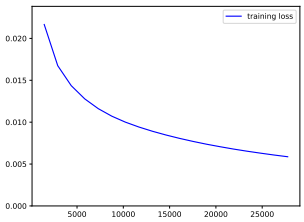
Figure 4 contains the analogous plots, but without $2/\eta$, and with the axis rescaled to zoom in on the SAM edge and the magnitudes of the principal eigenvalues of the Hessian. The operator norm closely tracks the SAM edge derived in Section 2. SAM operates at the edge of stability for a wider variety of learning rates than GD. We also see the SAM edge decreasing over time, as the gradients get smaller. The top three principal components are very close to one another. This is consistent with the view that SAM effectively performs gradient descent on the operator norm of the Hessian – if it did, a step would reduce the principal eigenvalue, while leaving the others at their old values, bringing the top eigenvalue closer to the others.

In Figure 5, we plot the training losses, when $\rho = 0.0$ and $\rho = 0.1$. SAM achieves flatter minima with similar loss. We also see that SAM drives training toward smoother regions in parameter space while the training error is still fairly high.

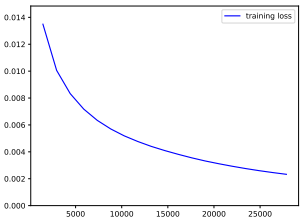
In Figure 6, we examine alignments between the gradients and the principal eigenvector of the Hessian, again where $\rho = 0.1$. We evaluate both the gradient at the iterate, and the gradient evaluated by SAM, at a distance ρ uphill. Since there are millions of parameters, random directions would have a tiny amount of alignment. We see a significant alignment between both gradients and the principal eigenvector of the Hessian, though the gradient used by SAM is aligned more closely. Recall that there are a number of eigenvectors whose



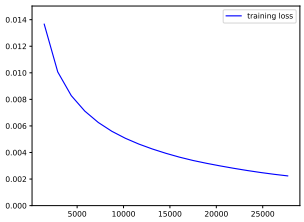
(a) $\eta = 0.03, \rho = 0$



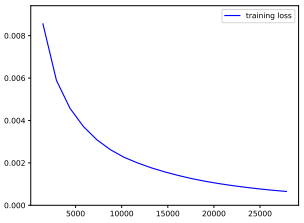
(b) $\eta = 0.03, \rho = 0.1$



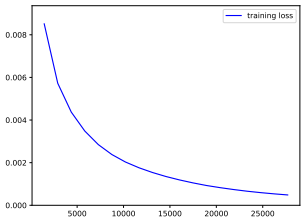
(c) $\eta = 0.1, \rho = 0$



(d) $\eta = 0.1, \rho = 0.1$

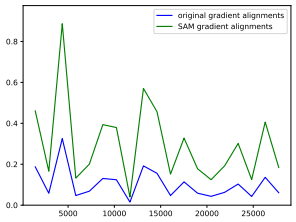


(e) $\eta = 0.3, \rho = 0$

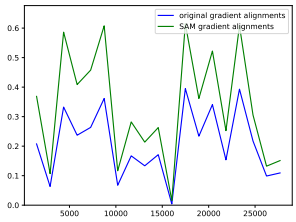


(f) $\eta = 0.3, \rho = 0.1$

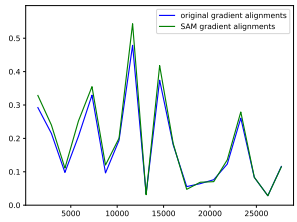
Figure 5: Training loss with GD and SAM on MNIST.



(a) $\eta = 0.03$

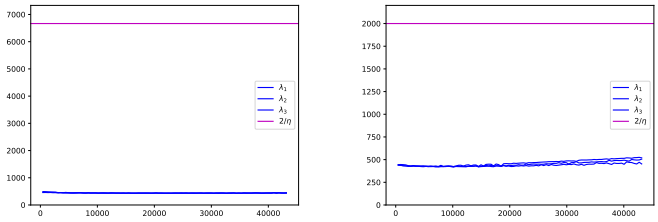


(b) $\eta = 0.1$

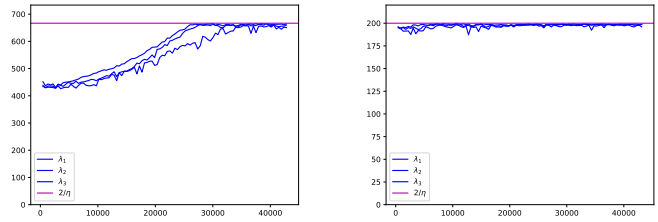


(c) $\eta = 0.3$

Figure 6: Alignments between gradients and the principal eigenvector of the Hessian with SAM on MNIST when $\rho = 0.1$.

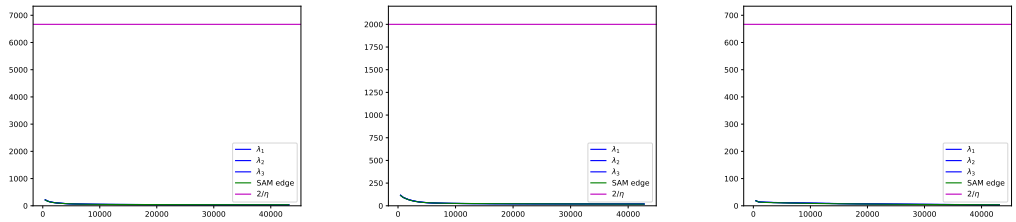


(a) $\eta = 0.0003$ (b) $\eta = 0.001$



(c) $\eta = 0.003$ (d) $\eta = 0.01$

Figure 7: Magnitudes of the largest eigenvalues of the Hessian when a CNN is trained with GD on 1000 examples from CIFAR10.



(a) $\eta = 0.0003$ (b) $\eta = 0.001$ (c) $\eta = 0.003$

Figure 8: Magnitudes of the largest eigenvalues of the Hessian when a CNN is trained with SAM, with $\rho = 0.1$, on CIFAR10.

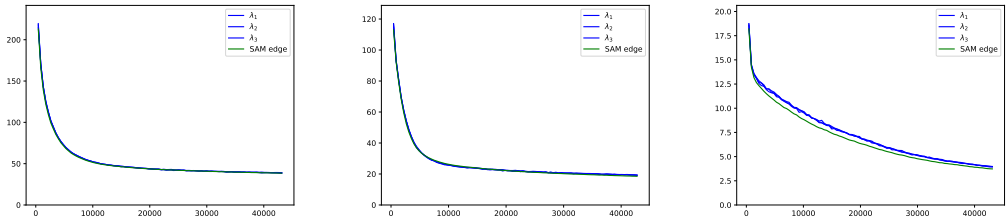
eigenvalues are nearly equal to the largest value. Reducing their eigenvalues can also make progress toward ultimately reducing the operator norm of the Hessian.

4.2 CIFAR10

In this section, we report on experiments with convolutional neural networks trained on 1000 examples from CIFAR10.

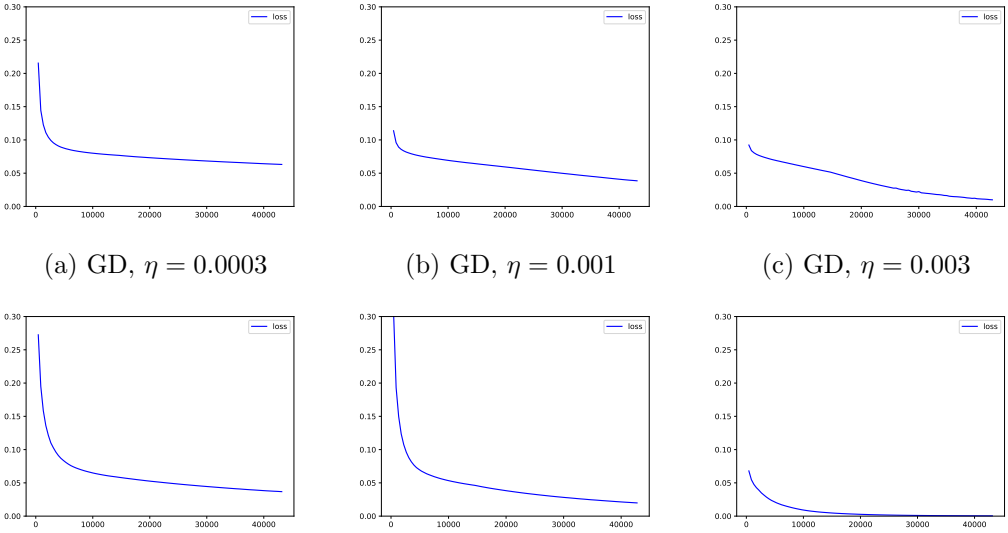
As before, we start with the case of GD in Figure 7. At the larger learning rates, training is reaching the edge of stability.

Next, we plot the same quantities when the network is trained with SAM, with $\rho = 0.1$, in Figure 8. Here, the eigenvalues are multiple orders of magnitude smaller than $2/\eta$.



(a) $\eta = 0.0003$ (b) $\eta = 0.001$ (c) $\eta = 0.003$

Figure 9: Magnitudes of the largest eigenvalues of the Hessian when a CNN is trained with SAM, with $\rho = 0.1$, on CIFAR10.



(a) GD, $\eta = 0.0003$ (b) GD, $\eta = 0.001$ (c) GD, $\eta = 0.003$

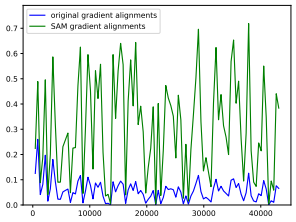
(d) SAM, $\eta = 0.0003$ (e) SAM, $\eta = 0.001$ (f) SAM, $\eta = 0.003$

Figure 10: Training loss with SGD and SAM (with $\rho = 0.1$) on CIFAR10.

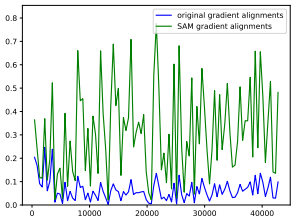
Next, in Figure 9 we no longer plot $2/\eta$, and zoom in on the region where the SAM edge and the eigenvalues are. Here, as with MNIST, we once again see SAM operating at the edge of stability identified in Section 2, even at learning rates where GD did not.

Figure 10 contains plots of the training loss on CIFAR10, for $\rho = 0.0$ and $\rho = 0.1$. In this task, SAM achieves wider minima without sacrificing training error. In fact, for the larger step sizes, its training error is better.

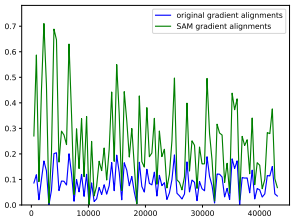
In Figure 11, we examine alignments between the gradients and the principal eigenvector of the Hessian in the case where $\rho = 0.1$ and a CNN is trained on CIFAR10. Again, we see significant alignment, especially at the higher learning rates. As in MNIST, we also see stronger alignment with the principal direction for the gradients evaluated at the uphill location used by SAM.



(a) $\eta = 0.0003$

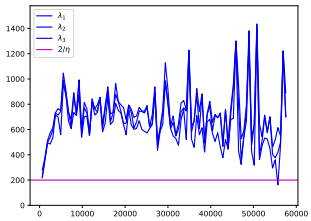


(b) $\eta = 0.001$

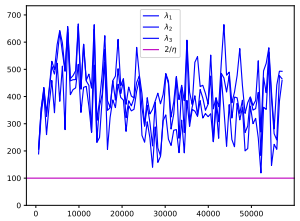


(c) $\eta = 0.003$

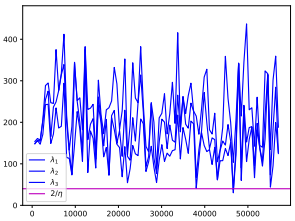
Figure 11: Alignments between gradients and the principal eigenvector of the Hessian with SAM on CIFAR10.



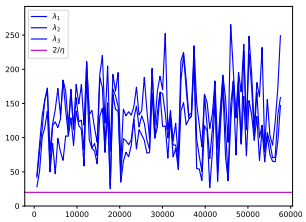
(a) $\eta = 0.01$



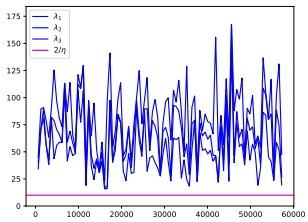
(b) $\eta = 0.02$



(c) $\eta = 0.05$



(d) $\eta = 0.1$



(e) $\eta = 0.2$

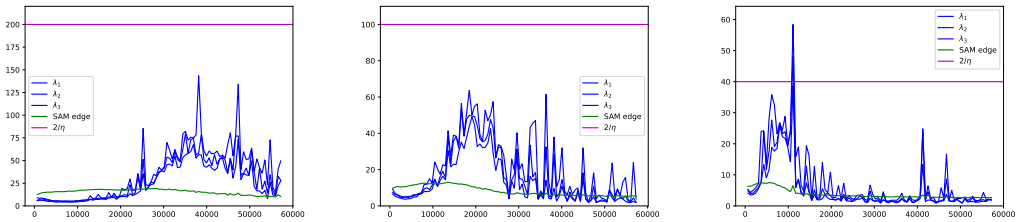
Figure 12: Magnitudes of the largest eigenvalues of the Hessian when a language model is trained with SGD.

4.3 Language modeling

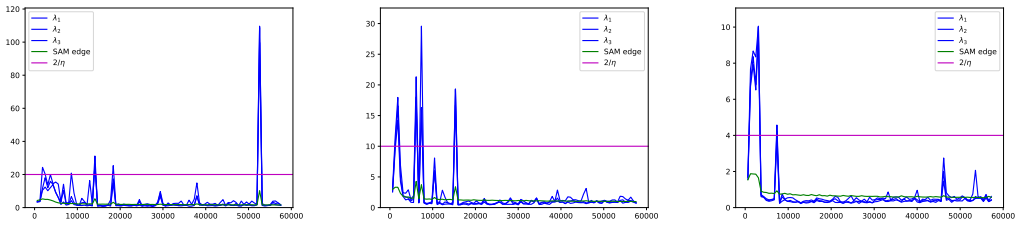
Next, we report on experiments training a language model. As before, we start with SGD, here in Figure 12.

Next, we plot the same quantities when the network is trained with SAM, with $\rho = 0.3$, in Figure 13. Here, the operator norm of the Hessian is significantly less than when SGD is used, and we see evidence that training in SAM operates at the edge of stability analyzed in Section 2. In Figure 14, we zoom in on the lower part of the curve, and plot the operator norm of the Hessian, to examine the relationship between this quantity and the SAM edge in more detail.

Figure 15 contains plots of the training loss, once again estimated per-minibatch. We included these mainly to motivate the combinations of hyperparameters where we examined

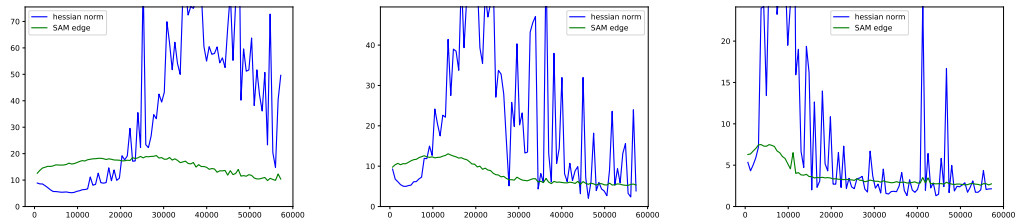


(a) $\eta = 0.01$ (b) $\eta = 0.02$ (c) $\eta = 0.05$

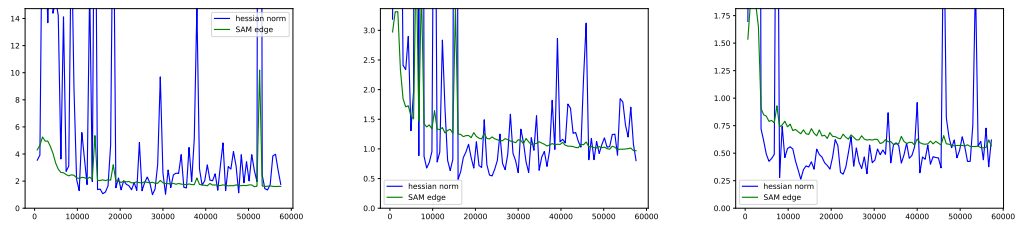


(d) $\eta = 0.1$ (e) $\eta = 0.2$ (f) $\eta = 0.5$

Figure 13: Magnitudes of the largest eigenvalues of the Hessian when a language model is trained with SAM, with $\rho = 0.3$.

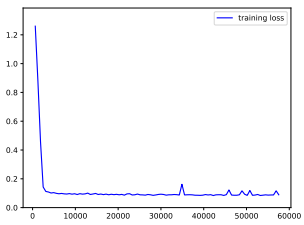


(a) $\eta = 0.01$ (b) $\eta = 0.02$ (c) $\eta = 0.05$

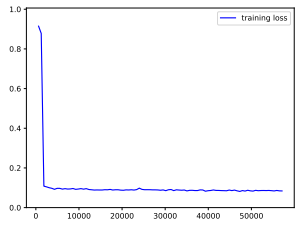


(d) $\eta = 0.1$ (e) $\eta = 0.2$ (f) $\eta = 0.5$

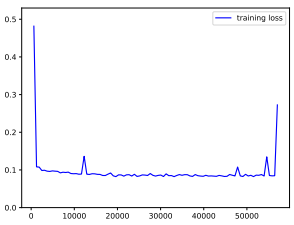
Figure 14: Magnitudes of the largest eigenvalues of the Hessian when a language model is trained with SAM, with $\rho = 0.3$.



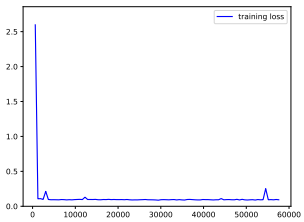
(a) $\rho = 0, \eta = 0.01$



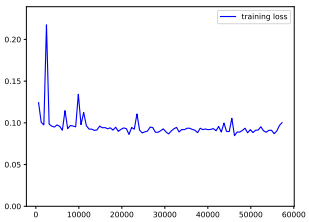
(b) $\rho = 0, \eta = 0.02$



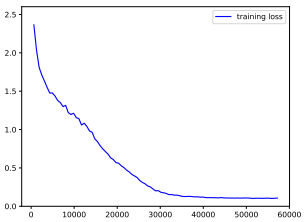
(c) $\rho = 0, \eta = 0.05$



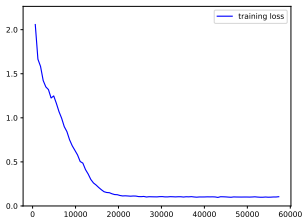
(d) $\rho = 0, \eta = 0.1$



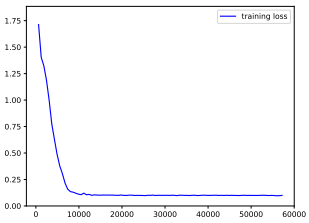
(e) $\rho = 0, \eta = 0.2$



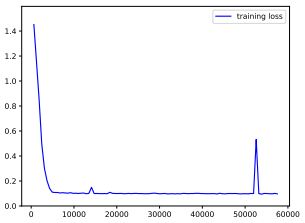
(f) $\rho = 0.3, \eta = 0.01$



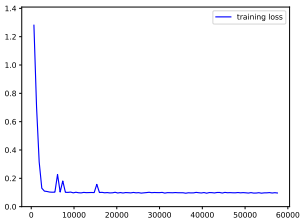
(g) $\rho = 0.3, \eta = 0.02$



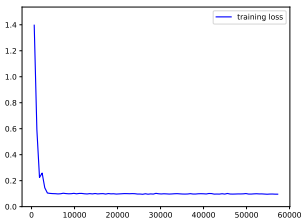
(h) $\rho = 0.3, \eta = 0.05$



(i) $\rho = 0.3, \eta = 0.1$

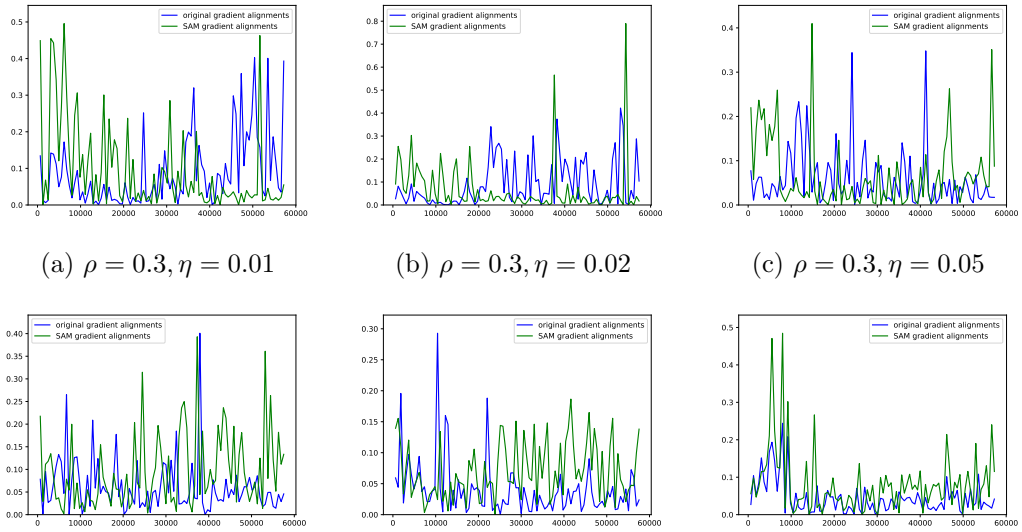


(j) $\rho = 0.3, \eta = 0.2$



(k) $\rho = 0.3, \eta = 0.5$

Figure 15: Training loss in the language modeling experiments.



(a) $\rho = 0.3, \eta = 0.01$ (b) $\rho = 0.3, \eta = 0.02$ (c) $\rho = 0.3, \eta = 0.05$

(d) $\rho = 0.3, \eta = 0.1$ (e) $\rho = 0.3, \eta = 0.2$ (f) $\rho = 0.3, \eta = 0.5$

Figure 16: Alignments between gradients and the principal direction of the Hessian in the language modeling experiments.

other aspects of the dynamics of SAM. As expected, while SAM does take longer to achieve a certain loss, it ultimately achieves training error similar to SGD, but with less sharpness.

Figure 16 contains plots of the alignment, once again estimated per-minibatch. For the large learning rates, late in training, despite the sampling noise arising from the use of minibatches, we see a systematic tendency for the SAM gradients to align more closely with the principal eigenvector of the Hessian than the gradients at the initial solution. However, for the smallest learning rates, the *opposite* holds.

5. Related work

In this section, we describe some previously mentioned related work in more detail, and also go over some additional papers.

Bartlett et al. (2023) analyzed the dynamics of SAM applied to a convex quadratic objective, and showed that it converges to oscillating in the direction of the principal eigenvector. Then they analyzed one step of SAM in more generality, starting at a solution near a local minimum, analogous to one of the steady-state solutions in the convex quadratic case. They showed that the update from this point can be decomposed into three terms, a term that corresponds to the update in the convex quadratic case (which moves to the other solution in the oscillation), a term in the descent direction of the operator norm of the Hessian, and a third term, which, for small η and ρ, is of lower order. The edge-of-stability point identified here is not a consequence of that analysis.

Among the varied results of Wen et al. (2023) is a theorem that may be paraphrased by saying that, for a smooth enough objective functions, in an overparameterized regime where there is a manifold of minimizers, once SAM's iterates are close to this manifold, its

updates track the updates that would be obtained by performing gradient flow to minimize the operator norm of the Hessian among minimizers of the loss. Their main results use the assumptions that $\eta \log(1/\rho)$ and ρ/η are sufficiently small. As was seen by Cohen et al. (2021) and also here, the edge-of-stability phenomenon dissipates as η gets small.

Andriushchenko et al. (2023) demonstrated empirically that networks trained by SAM tend to have features with lower rank, and illustrated how this can arise using a theoretical analysis of a two-layer network.

Cohen et al. (2022) demonstrated that some adaptive gradient methods, such as Adam, operate at the edge of stability.

A number of authors have provided insight by analyzing the dynamics of gradient descent under clean and simple conditions under which the edge of stability arises (see Zhu et al., 2022; Agarwala et al., 2023; Ahn et al., 2023; Chen and Bruna, 2022; Even et al., 2023). Properties of the loss landscape that are compatible with edge of stability training have also been described (and evaluated empirically) (Ma et al., 2022; Ahn et al., 2022). Arora et al. (2022) established conditions under which an algorithm like GD, but that normalizes the gradients so that they have unit length, operates at the edge of stability, and also analyzed an algorithm that takes gradients with respect to the square root of the loss.

Some authors have studied an algorithm like SAM, but, instead of updating using the gradient from the neighbor of the current iterate that is a constant distance ρ uphill, instead uses a gradient from neighbor whose distance from the current iterate scales with the norm of the gradient at the iterate (Andriushchenko and Flammarion, 2022; Agarwala and Dauphin, 2023), what has been called "unnormalized SAM". Dai et al. (2023) made a case that the SAM's normalization is crucial, motivating research into the original algorithm.

6. Conclusion

We have computed the critical value of operator norm of the Hessian corresponding to the edge of stability for SAM. This SAM-edge is a decreasing function of the norm of the gradient, so it tends to decrease as training progresses. For three deep learning training tasks, we have seen that the operator norm of the Hessian closely tracks this edge of stability, despite the noise introduced by estimating using minibatches in the `tiny_shakespeare` task.

SAM interacts strongly with the edge-of-stability phenomenon to drive down the operator norm of the Hessian, while also driving down the training error. Insight into how and why this happens could be promoted by identifying conditions under which SAM provably operates at its edge of stability, analogous to the results obtained for GD mentioned in Section 5. The analyses of Bartlett et al. (2023) and Wen et al. (2023) both required η and ρ to be small, and analyzed the effect of the dynamics on the operator norm of the Hessian late in training, whereas we empirically see a strong effect even early in training. One especially interesting question is how the training error is reduced so rapidly despite the overshooting associated with edge-of-stability training.

The experiments with language models showed that the edge-of-stability phenomenon can also be seen, to a limited extent, when training with SGD. A more thorough understanding of SAM and the edge of stability when training with SGD is another interesting and important subject for further research. (Wen et al. (2023) analyzed a variant of SAM that works using SGD one example at a time, and pointed out strong qualitative differences

between the algorithm that works with batch gradients and this extreme version of SGD, suggesting that interesting and rich structure might be found in the behavior of SAM with minibatches of intermediate size.)

In our experiments, there was a general tendency for the gradients used by SAM to be more aligned with the principal direction of the Hessian than gradients evaluated at the iterates. It is not clear why this is the case, and under what conditions it happens. The theoretical analysis by Bartlett et al. (2023) depended critically on the assumption that the update gradient was aligned with the principal eigenvector of the Hessian, which raises the possibility that the fact that the gradients used by SAM are aligned more closely with the principal direction of the Hessian is key to its success. However, it is not clear under what conditions, and why, this improved alignment is seen, and when it is helpful. There also was an intriguing exception when language models were trained with SGD using small step sizes that it would be interesting to further explore.

Acknowledgments

We thank Naman Agarwal and Hossein Mobahi for valuable conversations, and Naman Agarwal for his comments on an earlier version of this paper. PB gratefully acknowledges the support of the NSF through grants DMS-2023505 and DMS-2031883 and of Simons Foundation award #814639.

References

Atish Agarwala and Yann Dauphin. SAM operates far from home: eigenvalue regularization as a dynamical phenomenon. In *International Conference on Machine Learning*, pages 152–168, 2023.

Atish Agarwala, Fabian Pedregosa, and Jeffrey Pennington. Second-order regression models exhibit progressive sharpening to the edge of stability. In *ICML*, volume 202, pages 169–195. PMLR, 2023.

Kwangjun Ahn, Jingzhao Zhang, and Suvrit Sra. Understanding the unstable convergence of gradient descent. In *International Conference on Machine Learning*, pages 247–257. PMLR, 2022.

Kwangjun Ahn, Sébastien Bubeck, Sinho Chewi, Yin Tat Lee, Felipe Suarez, and Yi Zhang. Learning threshold neurons via the "edge of stability". *NeurIPS*, 2023.

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, pages 639–668, 2022.

Maksym Andriushchenko, Dara Bahri, Hossein Mobahi, and Nicolas Flammarion. Sharpness-aware minimization leads to low-rank features. *NeurIPS*, 2023.

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In *International Conference on Machine Learning*, pages 948–1024, 2022.

Peter L Bartlett, Philip M Long, and Olivier Bousquet. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *JMLR*, 24 (316):1–39, 2023.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`. Version 0.3.13.

Lei Chen and Joan Bruna. On gradient descent convergence beyond the edge of stability. *arXiv preprint arXiv:2206.04172*, 2022.

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *ICLR*, 2021.

Jeremy M Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E Dahl, et al. Adaptive gradient methods at the edge of stability. *arXiv preprint arXiv:2207.14484*, 2022.

Yan Dai, Kwangjun Ahn, and Suvrit Sra. The crucial role of normalization in sharpness-aware minimization. *NeurIPS*, 2023.

Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. *ICLR*, 2023.

Mathieu Even, Scott Pesme, Suriya Gunasekar, and Nicolas Flammarion. (S) GD over diagonal linear networks: Implicit regularisation, large stepsizes and edge of stability. *NeurIPS*, 2023.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241. PMLR, 2019.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2023. URL `http://github.com/google/flax`. Version 0.7.2.

Tom Hennigan, Trevor Cai, Tamara Norman, Lena Martens, and Igor Babuschkin. Haiku: Sonnet for JAX, 2020. URL `http://github.com/deepmind/dm-haiku`. Version 0.0.10.

Tom Hennigan, Trevor Cai, Tamara Norman, Lena Martens, and Igor Babuschkin, 2023. URL `https://github.com/deepmind/dm-haiku/blob/main/examples/transformer/train.py`. Downloaded 9/1/23.

Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2019.

P. M. Long and P. L. Bartlett. Sam and the edge of stability. `https://github.com/google-deepmind/sam_edge`, 2024.

C. Ma, D. Kunin, L. Wu, and L. Ying. Beyond the quadratic approximation: The multiscale structure of neural network loss landscapes. *Journal of Machine Learning*, 1(3):247–267, 2022.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How sharpness-aware minimization minimizes sharpness? In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=5spDgWmpY6x`.

Xingyu Zhu, Zixuan Wang, Xiang Wang, Mo Zhou, and Rong Ge. Understanding edge-of-stability training dynamics with a minimalist example. In *The Eleventh International Conference on Learning Representations*, 2022.