“Sharp Minima Can Generalize For Deep Nets,”

L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio

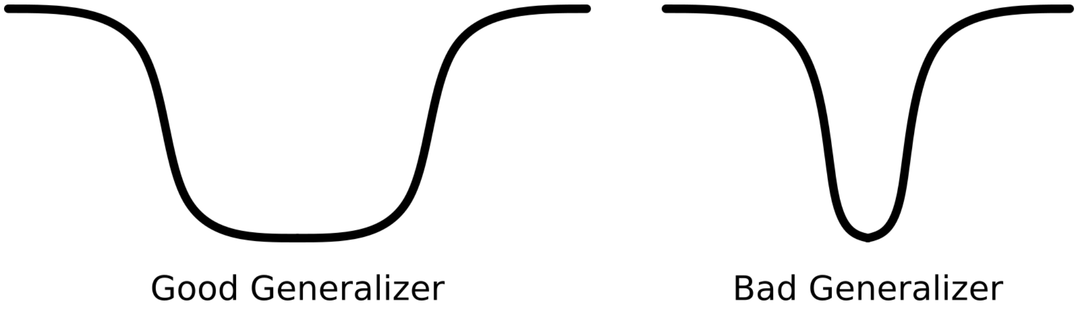
Mar. 2017.

At a loss: Landscaping loss, and how it affects generalization
-Adam

Flat Minima Generalize Well

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima,” *arXiv:1609.04836 [cs, math]*, Sep. 2016.

“We investigate the cause for this generalization drop in the large-batch regime and present numerical evidence that supports the view that large-batch methods tend to converge to sharp minimizers of the training and testing functions—**and as is well known, sharp minima lead to poorer generalization**.”



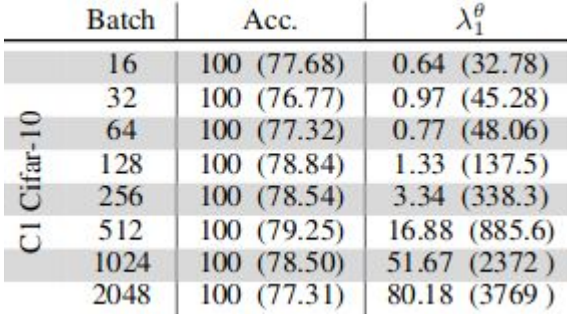
Flat Minima Can Generalize Well

When using stochastic gradient descent:

- batches of size 256 generalize better and are flatter than batches of size 10%

Flat Minima Generalize Well?

Is it really flat minima that generalize well?



	Batch	Acc.	λ_1^θ
C1 Cifar-10	16	100 (77.68)	0.64 (32.78)
	32	100 (76.77)	0.97 (45.28)
	64	100 (77.32)	0.77 (48.06)
	128	100 (78.84)	1.33 (137.5)
	256	100 (78.54)	3.34 (338.3)
	512	100 (79.25)	16.88 (885.6)
	1024	100 (78.50)	51.67 (2372)
	2048	100 (77.31)	80.18 (3769)

Z. Yao, A. Gholami, Q. Lei, K. Keutzer, and M. W. Mahoney, “Hessian-based Analysis of Large Batch Training and Robustness to Adversaries,” arXiv:1802.08241 [cs, stat], Feb. 2018.

Why do flat minima generalize well?

Presuming that flat minima mark good generalization

Which of the following are true:

1. Flat minima are the reason for good generalization
2. Gradient descent methods gravitate to flat minima more easily.
    a. Thus models that generalize get chosen over models that do not, causing the model selection process to biased towards performing well in flat minima for the data regimes that they have been selected for.
3. Something else

Do flat minima mark good generalization?

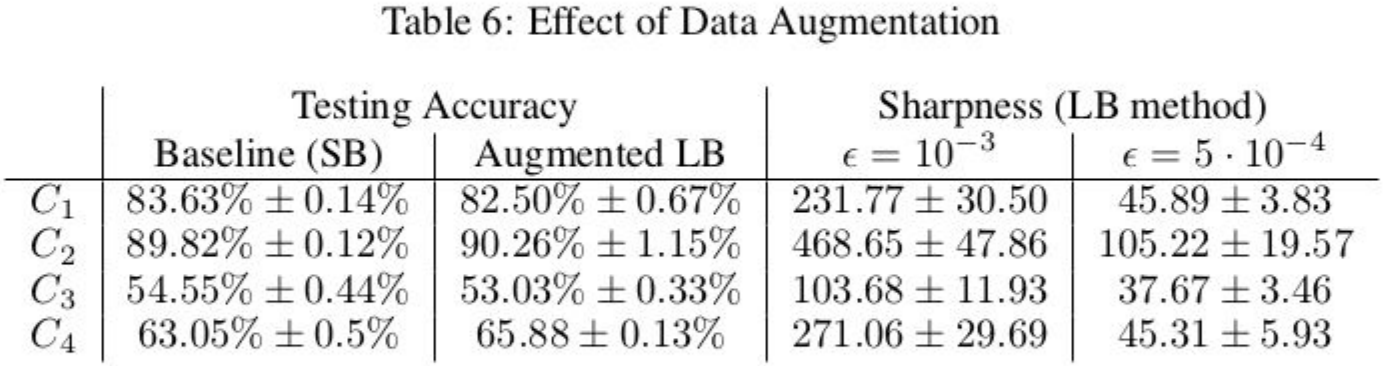
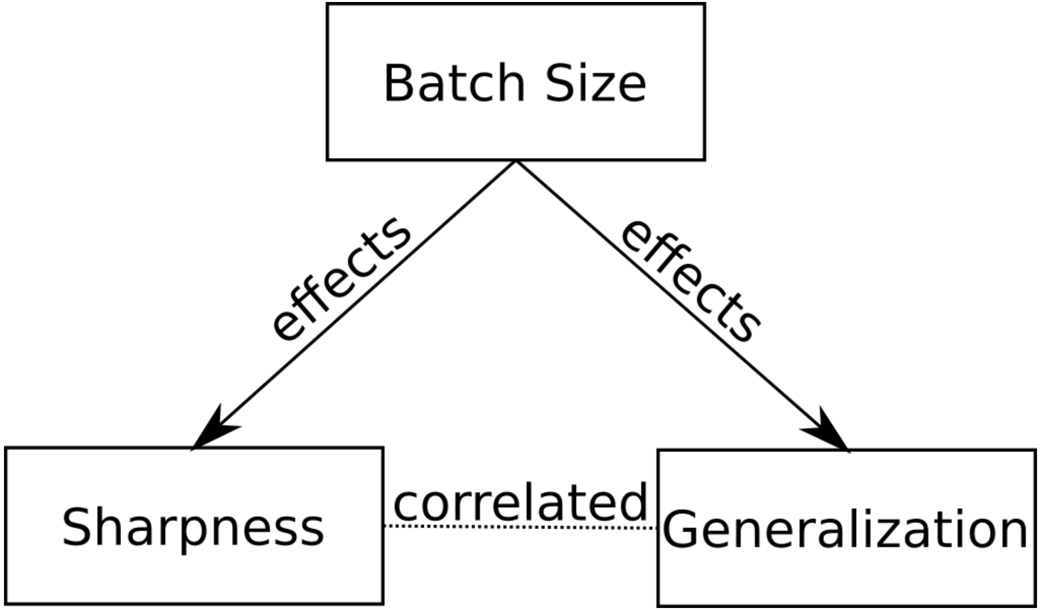


Table 6: Effect of Data Augmentation

	Testing Accuracy		Sharpness (LB method)	
	Baseline (SB)	Augmented LB	$\epsilon = 10^{-3}$	$\epsilon = 5 \cdot 10^{-4}$
C_1	$83.63\% \pm 0.14\%$	$82.50\% \pm 0.67\%$	231.77 ± 30.50	45.89 ± 3.83
C_2	$89.82\% \pm 0.12\%$	$90.26\% \pm 1.15\%$	468.65 ± 47.86	105.22 ± 19.57
C_3	$54.55\% \pm 0.44\%$	$53.03\% \pm 0.33\%$	103.68 ± 11.93	37.67 ± 3.46
C_4	$63.05\% \pm 0.5\%$	$65.88 \pm 0.13\%$	271.06 ± 29.69	45.31 ± 5.93

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima,” *arXiv:1609.04836 [cs, math]*, Sep. 2016.

Coarse Model for Minima



“Sharp Minima Can Generalize For Deep Nets,”

L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio

Mar. 2017.

Broad Strokes

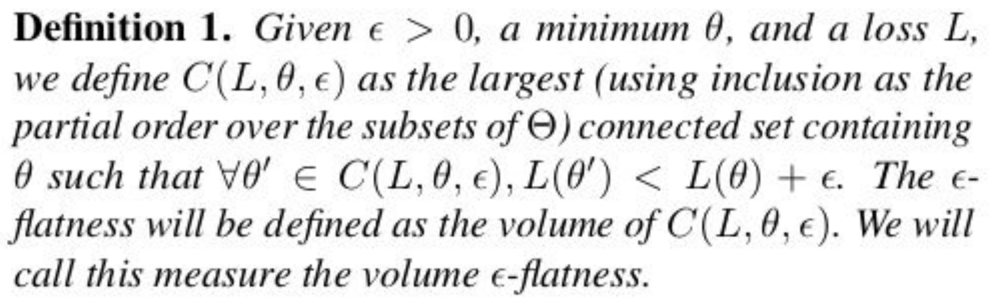
Models can be tweaked after training to adjust sharpness according to common metrics.

Common Flatness Metrics

1. Volume flatness
2. ϵ-sharpness
3. Hessian Based

Flatness 1

Volume flatness:

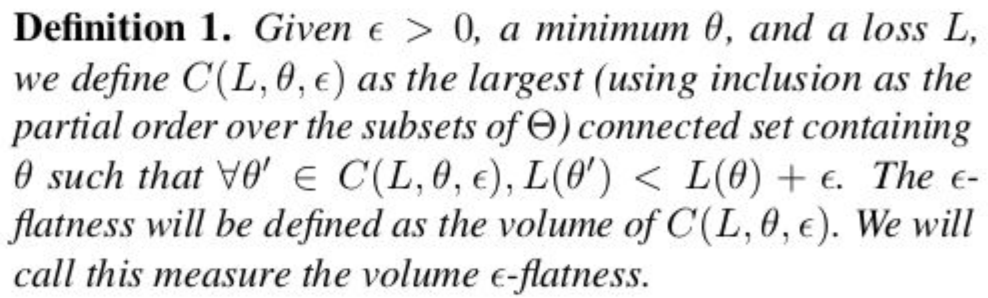


Definition 1. *Given $\epsilon > 0$, a minimum θ, and a loss L, we define $C(L, \theta, \epsilon)$ as the largest (using inclusion as the partial order over the subsets of Θ) connected set containing θ such that $\forall \theta' \in C(L, \theta, \epsilon), L(\theta') < L(\theta) + \epsilon$. The ϵ-flatness will be defined as the volume of $C(L, \theta, \epsilon)$. We will call this measure the volume ϵ-flatness.*

“Sharp Minima Can Generalize For Deep Nets,” L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio

Flatness 1

Volume flatness has infinite volume for rectified neural networks.

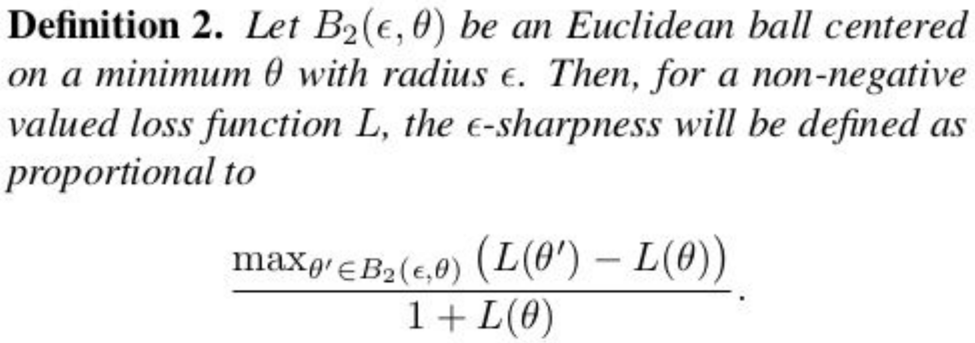


Definition 1. *Given $\epsilon > 0$, a minimum θ, and a loss L, we define $C(L, \theta, \epsilon)$ as the largest (using inclusion as the partial order over the subsets of Θ) connected set containing θ such that $\forall \theta' \in C(L, \theta, \epsilon), L(\theta') < L(\theta) + \epsilon$. The ϵ-flatness will be defined as the volume of $C(L, \theta, \epsilon)$. We will call this measure the volume ϵ-flatness.*

“Sharp Minima Can Generalize For Deep Nets,” L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio

Flatness 2

ϵ-sharpness: (Keskar et. al flatness)



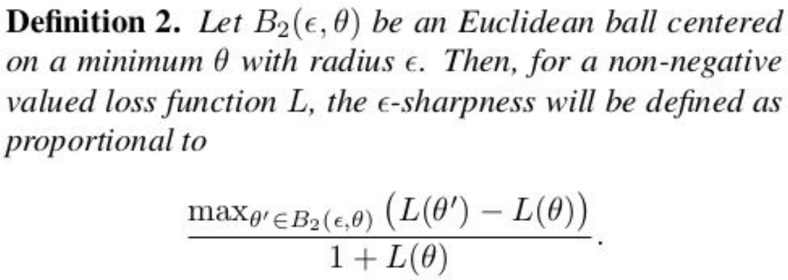
Definition 2. *Let $B_2(\epsilon, \theta)$ be an Euclidean ball centered on a minimum θ with radius ϵ. Then, for a non-negative valued loss function L, the ϵ-sharpness will be defined as proportional to*

$$\frac{\max_{\theta' \in B_2(\epsilon,\theta)} \left(L(\theta') - L(\theta)\right)}{1 + L(\theta)}.$$

“Sharp Minima Can Generalize For Deep Nets,” L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio

Flatness 2

ϵ-sharpness: (Keskar et. al flatness)



Definition 2. *Let $B_2(\epsilon, \theta)$ be an Euclidean ball centered on a minimum θ with radius ϵ. Then, for a non-negative valued loss function L, the ϵ-sharpness will be defined as proportional to*

$$\frac{\max_{\theta' \in B_2(\epsilon,\theta)} \left(L(\theta') - L(\theta)\right)}{1 + L(\theta)}.$$

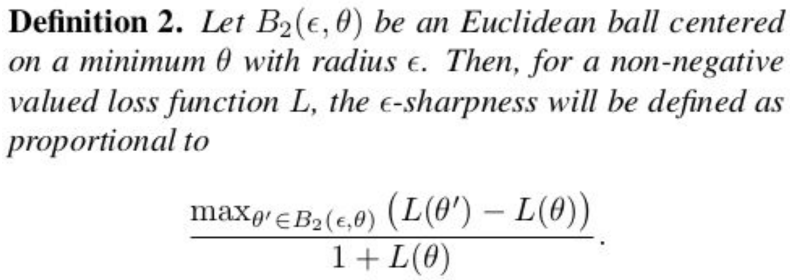
“Sharp Minima Can Generalize For Deep Nets,” L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio

“For rectified neural network every minimum is observationally equivalent to a minimum that generalizes as well but with high ϵ-sharpness. This also applies when using full-space ϵ-sharpness used by Keskar et al. (2017).”

Full-space is related to spectral norm.

Flatness 2

ϵ-sharpness: (Keskar et. al flatness)



Definition 2. *Let $B_2(\epsilon, \theta)$ be an Euclidean ball centered on a minimum θ with radius ϵ. Then, for a non-negative valued loss function L, the ϵ-sharpness will be defined as proportional to*

$$\frac{\max_{\theta' \in B_2(\epsilon,\theta)} \left(L(\theta') - L(\theta)\right)}{1 + L(\theta)}.$$

“Sharp Minima Can Generalize For Deep Nets,” L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio

“For rectified neural network every minimum is observationally equivalent to a minimum that generalizes as well but with high ϵ-sharpness. This also applies when using full-space ϵ-sharpness used by Keskar et al. (2017).”

Full-space is related to spectral norm of the Hessian.

What about other eigenvalues?

Flatness 2

What about other eigenvalues?

“We have not been able to show a similar problem with random subspace ϵ-sharpness used by Keskar et al. (2017)”

Hessian Flatness

Looks at network curvature is with respect to parameters

Hessian Flatness Quantification

Values used:

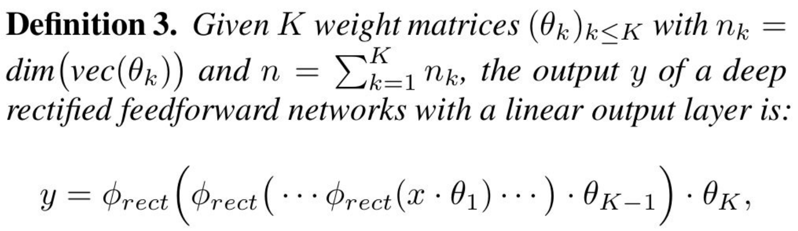
- Spectral Norm
- Trace

Hessian Flatness Quantification

Values used:

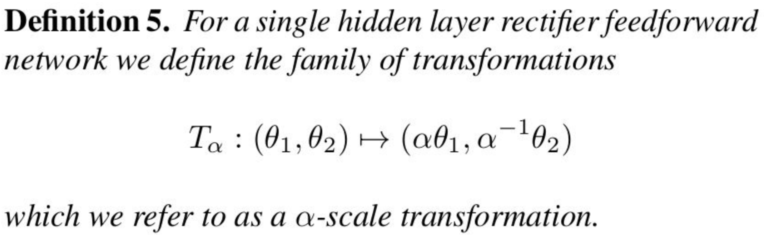
- Spectral Norm
- Trace (lower bounded by spectral norm)

Alpha-Scale Transformation



Definition 3. *Given K weight matrices $(\theta_k)_{k \leq K}$ with $n_k = dim\big(vec(\theta_k)\big)$ and $n = \sum_{k=1}^{K} n_k$, the output y of a deep rectified feedforward networks with a linear output layer is:*

$$y = \phi_{rect}\Big(\phi_{rect}\big(\cdots\phi_{rect}(x \cdot \theta_1)\cdots\big) \cdot \theta_{K-1}\Big) \cdot \theta_K,$$



Definition 5. *For a single hidden layer rectifier feedforward network we define the family of transformations*

$$T_\alpha : (\theta_1, \theta_2) \mapsto (\alpha\theta_1, \alpha^{-1}\theta_2)$$

which we refer to as a α-scale transformation.

2-layer version

$$
\begin{aligned}
&(\nabla^2 L)(\alpha\theta_1, \alpha^{-1}\theta_2) \\
&= \begin{bmatrix} \alpha^{-1}\mathbb{I}_{n_1} & 0 \\ 0 & \alpha\mathbb{I}_{n_2} \end{bmatrix} (\nabla^2 L)(\theta_1, \theta_2) \begin{bmatrix} \alpha^{-1}\mathbb{I}_{n_1} & 0 \\ 0 & \alpha\mathbb{I}_{n_2} \end{bmatrix}.
\end{aligned}
$$

Spectral norm can be arbitrarily scaled

Multi-layer version

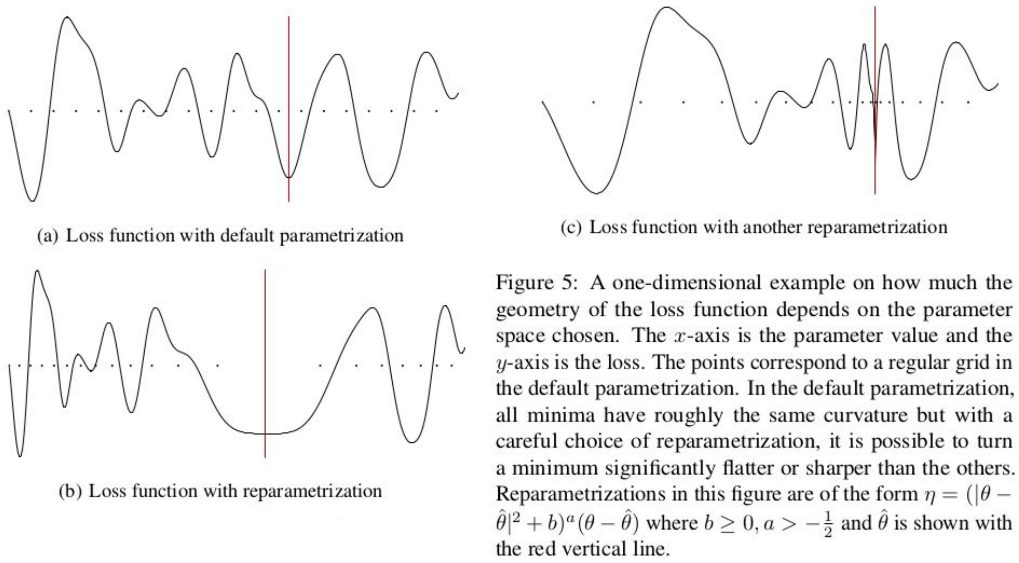
Eigenvalues of n-1 layers can be scaled arbitrarily large, where n is the count of the layers.

Hessian Measures that can avoid this

Maybe a product of hessian eigenvalues… can still be scaled to be sharper, but relative sharpness will be maintained.

Model re-parameterization & input space

Can re-parameterize weights as function to change landscape.



(a) Loss function with default parametrization

(c) Loss function with another reparametrization

(b) Loss function with reparametrization

Figure 5: A one-dimensional example on how much the geometry of the loss function depends on the parameter space chosen. The x-axis is the parameter value and the y-axis is the loss. The points correspond to a regular grid in the default parametrization. In the default parametrization, all minima have roughly the same curvature but with a careful choice of reparametrization, it is possible to turn a minimum significantly flatter or sharper than the others. Reparametrizations in this figure are of the form $\eta = (|\theta - \hat{\theta}|^2 + b)^a(\theta - \hat{\theta})$ where $b \geq 0, a > -\frac{1}{2}$ and $\hat{\theta}$ is shown with the red vertical line.

Conclusion

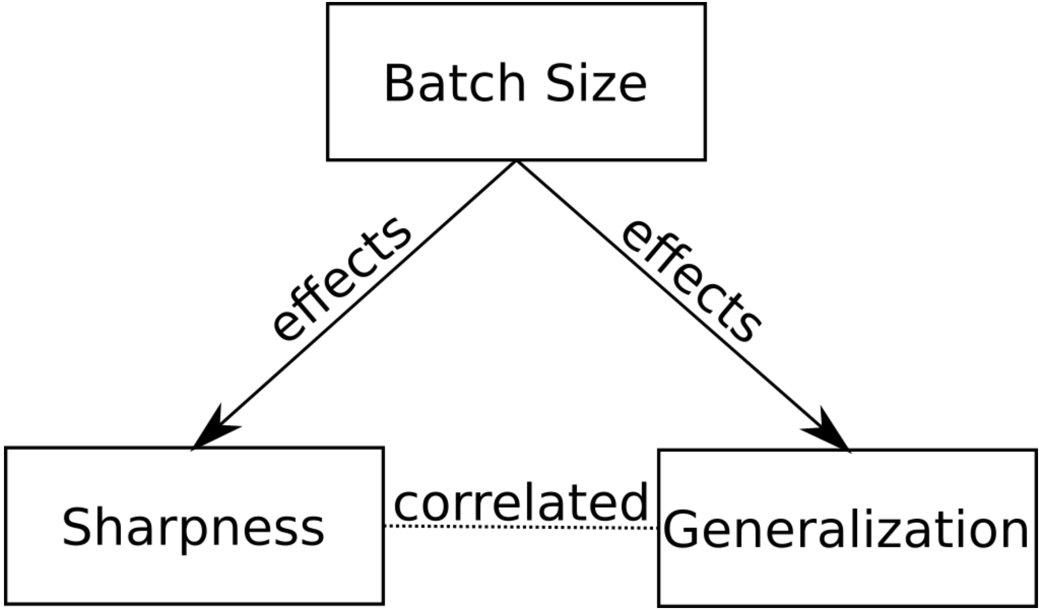
Flat minima don’t necessarily generalize better than sharp ones.

Exploiting non-identifiability allows changing surface without affecting function.

Conclusion

Flat minima don’t necessarily generalize better than sharp ones.

Exploiting non-identifiability allows changing surface without affecting function.

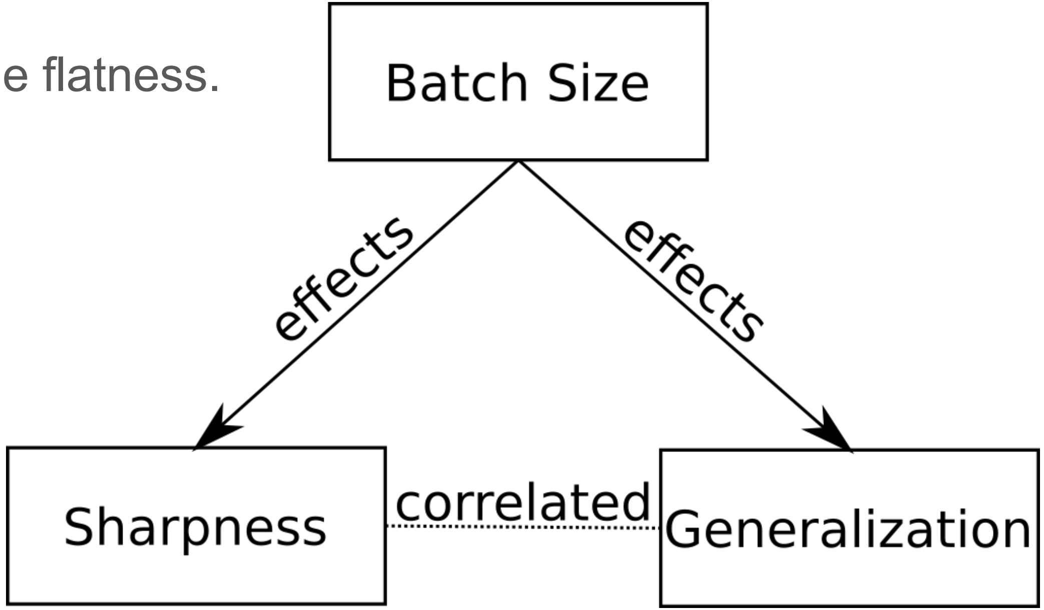


Conclusion

Flat minima don’t necessarily generalize better than sharp ones.

Exploiting non-identifiability allows changing surface without affecting function.

More care is needed to define flatness.



Bayesian method

S. L. Smith and Q. V. Le, “A Bayesian Perspective on Generalization and Stochastic Gradient Descent,” arXiv:1710.06451 [cs, stat], Oct. 2017.

Likely models are a combination of depth and breadth

$$P(y|x;M) \approx \lambda^{\frac{p}{2}} e^{-C(\omega_0)} / |\nabla\nabla C(\omega)|_{\omega_0}^{1/2}$$

$$(\lambda^{\frac{p}{2}} / |\nabla\nabla C(\omega)|_{\omega_0}^{1/2})$$

Occam Factor

Regularization constant divided by product of eigenvalues of hessian.

$$(\lambda^{\frac{p}{2}} / |\nabla \nabla C(\omega)|_{\omega_0}^{1/2})$$

SGD Noise Scale

S. L. Smith and Q. V. Le, “A Bayesian Perspective on Generalization and Stochastic Gradient Descent,” arXiv:1710.06451 [cs, stat], Oct. 2017.

$$g = \epsilon(\frac{N}{B} - 1) \approx \epsilon N/B$$

$$g = \epsilon\left(\frac{N}{B} - 1\right) \approx \epsilon N/B$$

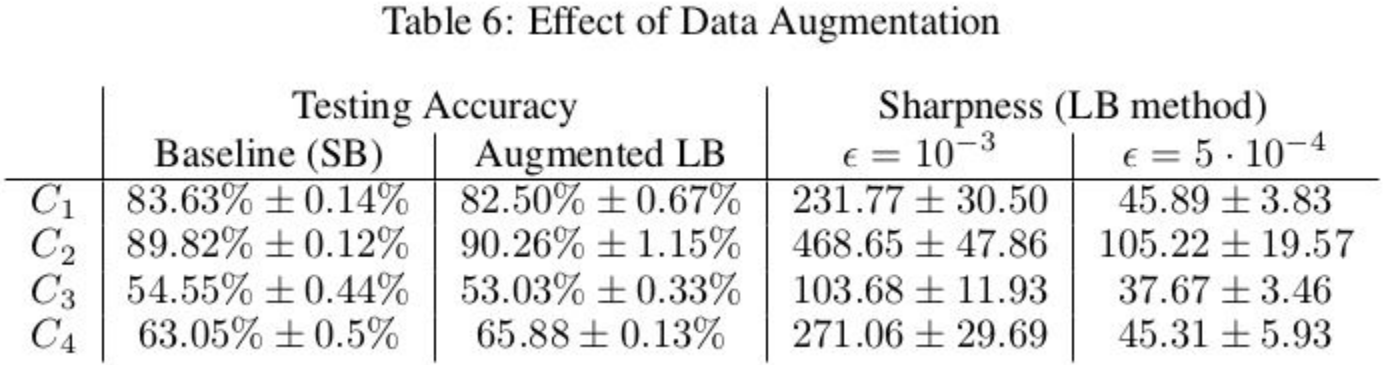


Table 6: Effect of Data Augmentation

	Testing Accuracy		Sharpness (LB method)	
	Baseline (SB)	Augmented LB	$\epsilon = 10^{-3}$	$\epsilon = 5 \cdot 10^{-4}$
C_1	$83.63\% \pm 0.14\%$	$82.50\% \pm 0.67\%$	231.77 ± 30.50	45.89 ± 3.83
C_2	$89.82\% \pm 0.12\%$	$90.26\% \pm 1.15\%$	468.65 ± 47.86	105.22 ± 19.57
C_3	$54.55\% \pm 0.44\%$	$53.03\% \pm 0.33\%$	103.68 ± 11.93	37.67 ± 3.46
C_4	$63.05\% \pm 0.5\%$	$65.88 \pm 0.13\%$	271.06 ± 29.69	45.31 ± 5.93

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima,” *arXiv:1609.04836 [cs, math]*, Sep. 2016.

Addendum

Warm-starting models...

Check out: S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le, “Don’t Decay the Learning Rate, Increase the Batch Size,” *arXiv:1711.00489 [cs, stat]*, Nov. 2017.

Is flatness a goose-chase?

Why focus on flatness rather than directly on training paradigms that directly affect generalization?

Can we optimize over the model itself to improve flatness as well? Eg. Skip-connections, other online transformations