On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal,
Mikhail Smelyanskiy, Ping Tak Peter Tang

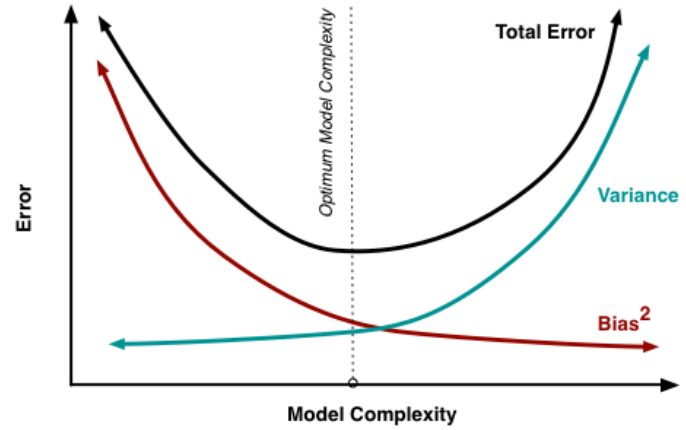
Intel Corporation

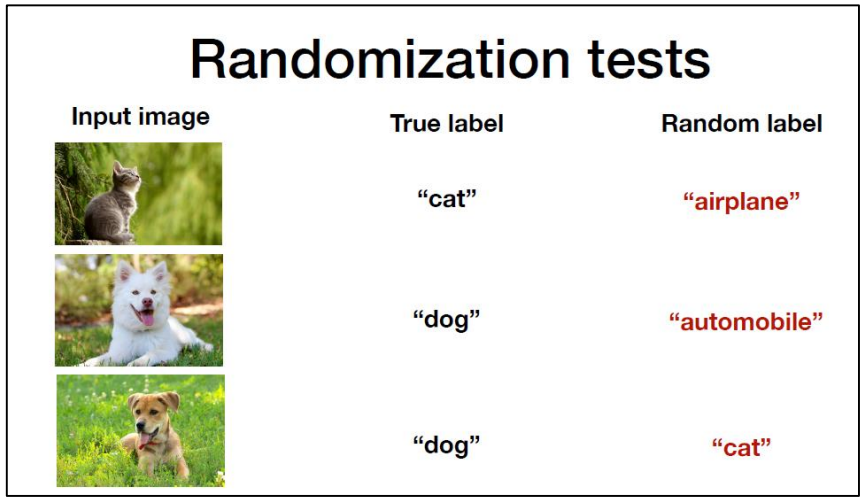
ICLR 2017

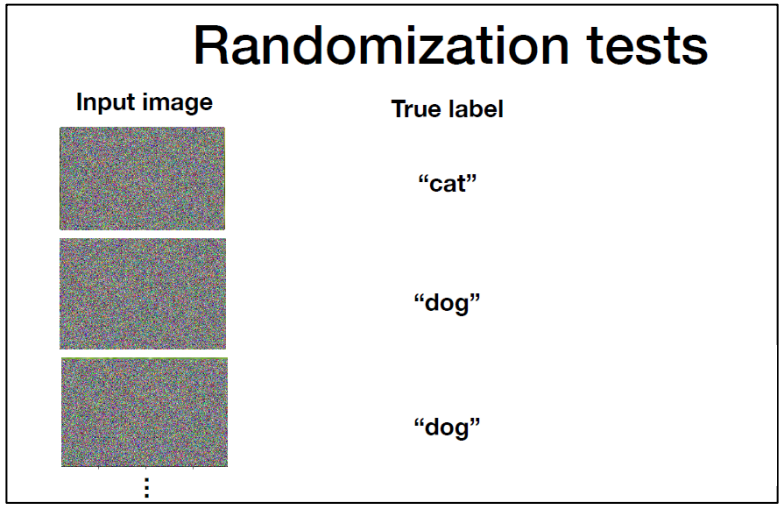
Presented by Amir Abdi
amirabdi@ece.ubc.ca
Oct 10, 2019

What happened on MLRG

- Overparameterized deep networks can easily **fit random labels**
- Overparameterized neural nets lead to generalization bounds
- Regularization helps with test set generalization performance but doesn't affect generalization bounds







Presentation by Jason Hartford, October 2

Understanding deep learning requires rethinking generalization, Zhang et al.

Key contributions

- Empirical study on the drawback of **large-batch (LB)** methods vs **small-batch (SB)**
- Empirical study of **sharpness** of minimizers
- **Generalization gap** is correlated with **sharpness** of minimizers
- Some attempts to improve performance of large-batch

SGD

- Theoretical guarantee
  - Convergence to minimizers of strongly-convex functions and to stationary points for non-convex functions (Bottou et al., 2016)
  - Saddle-point avoidance (Ge et al., 2015; Lee et al., 2016)

- The **sequential nature** of SGD, limits parallelization.
- Increasing batch-size would improve parallelization (assuming enough processing cores), but degrades performance on test-test (generalization gap)

Notations

Non-convex optimization

$$\min_{x \in \mathbb{R}^n} \quad f(x) := \frac{1}{M} \sum_{i=1}^{M} f_i(x)$$

f_i : loss function for data point i

x is the vector of weights being optimized over each iteration using SGD

$$x_{k+1} = x_k - \alpha_k \left(\frac{1}{|B_k|} \sum_{i \in B_k} \nabla f_i(x_k) \right)$$

α	step size
$\|B_k\|$	batch size at iteration k
M	dataset size

Empirical Settings

- Dataset

Data Set	# Data Points Train	Test	# Features	# Classes
MNIST	60000	10000	28×28	10
TIMIT	721329	310621	360	1973
CIFAR-10	50000	10000	32×32	10
CIFAR-100	50000	10000	32×32	100

- Network Architectures
  - F1: 5 hidden layers
  - F2: 7 hidden layers
  - C1 and C3: AlexNet
  - C2 and C4: VGG

Name	Network Type	Data set
F_1	Fully Connected	MNIST
F_2	Fully Connected	TIMIT
C_1	(Shallow) Convolutional	CIFAR-10
C_2	(Deep) Convolutional	CIFAR-10
C_3	(Shallow) Convolutional	CIFAR-100
C_4	(Deep) Convolutional	CIFAR-100

- Batch-size
  - Small-batch (SB): $|B| = 256$
  - Large-batch (LB): $|B| = M/10$
- ADAM as the SGD optimizer
- Softmax with cross-entropy loss
- All experiments are repeated 5 times (mean ± std reported)

Results

- All experiments repeated 5 times from uniformly distributed random starting points

	Training Accuracy		Testing Accuracy	
Name	SB	LB	SB	LB
F_1	$99.66\% \pm 0.05\%$	$99.92\% \pm 0.01\%$	$98.03\% \pm 0.07\%$	$97.81\% \pm 0.07\%$
F_2	$99.99\% \pm 0.03\%$	$98.35\% \pm 2.08\%$	$64.02\% \pm 0.2\%$	$59.45\% \pm 1.05\%$
C_1	$99.89\% \pm 0.02\%$	$99.66\% \pm 0.2\%$	$80.04\% \pm 0.12\%$	$77.26\% \pm 0.42\%$
C_2	$99.99\% \pm 0.04\%$	$99.99\% \pm 0.01\%$	$89.24\% \pm 0.12\%$	$87.26\% \pm 0.07\%$
C_3	$99.56\% \pm 0.44\%$	$99.88\% \pm 0.30\%$	$49.58\% \pm 0.39\%$	$46.45\% \pm 0.43\%$
C_4	$99.10\% \pm 1.23\%$	$99.57\% \pm 1.84\%$	$63.08\% \pm 0.5\%$	$57.81\% \pm 0.17\%$

No statistically sig diff in training

Up to 5% drop in performance
(Generalization gap)

Conjectures

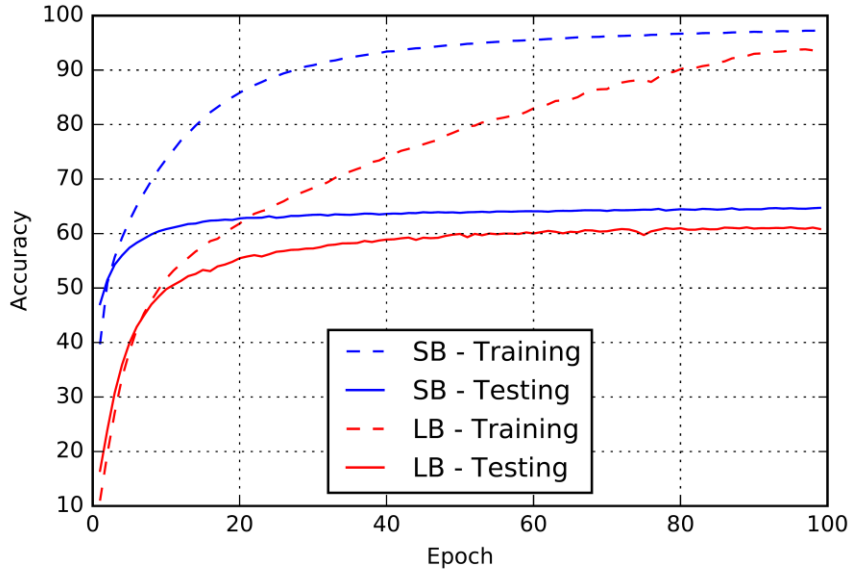
1. LB methods **over-fit** the model

2. LB methods are attracted to **saddle points**

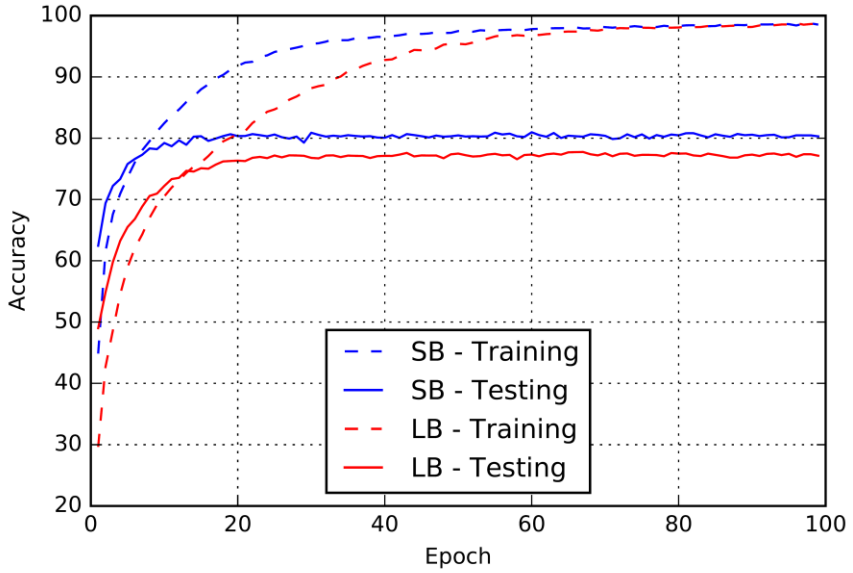
3. LB methods lack the **explorative** properties of SB methods and tend to zoom-in on the minimizer closest to the initial point

SB and LB methods converge to **qualitatively different minimizers** with differing generalization properties.

Observation: No overfitting



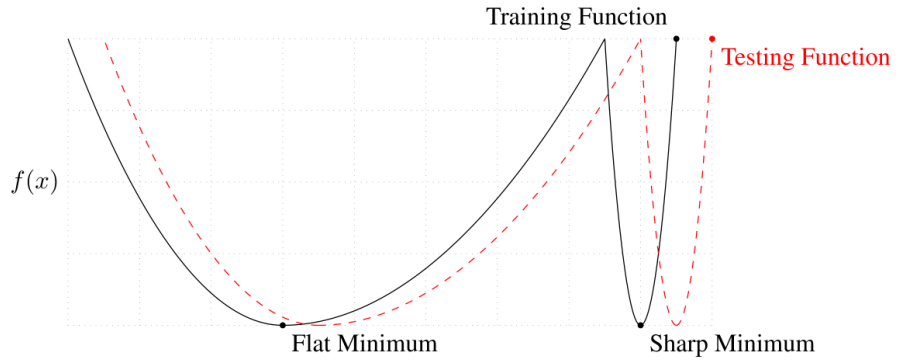
(a) Network F_2



(b) Network C_1

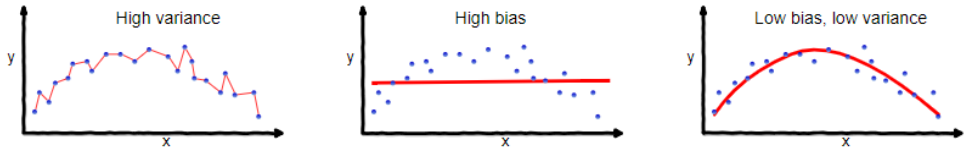
Sharpness of Minimizers

- Flat minimizer: the function varies slowly in a relatively large neighborhood of $\bar{x}$
- Sharp minimizer: the function increases rapidly in a relatively small neighborhood of $\hat{x}$



✓ The large **sensitivity** of the training function at a sharp minimizer negatively impacts the ability of the trained model to **generalize** on new data

✓ Minimum Description Length (MDL): Statistical models that are **less complex** (can be described with less precision)**, generalize better** (Rissanen, 1983).

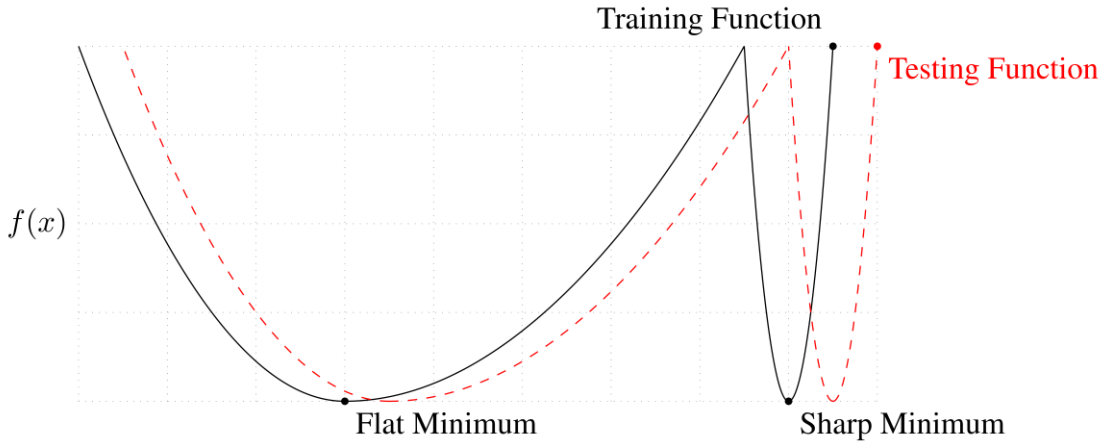


overfitting underfitting Good balance

Hochreiter and Schmidhuber. Flat minima. Neural Computation. 1997.

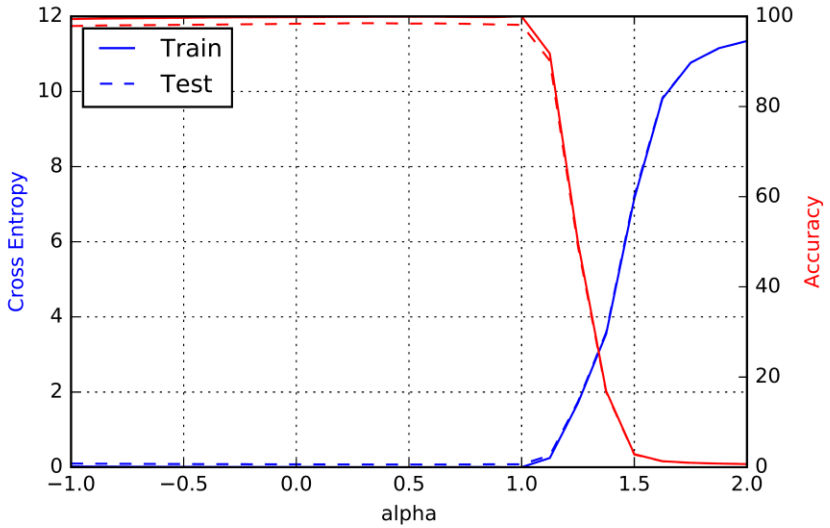
Main observation

The lack of generalization ability is due to the fact that large-batch methods tend to converge to *sharp minimizers* of the training function. These minimizers are characterized by a significant number of large positive eigenvalues in $\nabla^2 f(x)$, and tend to generalize less well. In contrast, small-batch methods converge to *flat minimizers* characterized by having numerous small eigenvalues of $\nabla^2 f(x)$. We have observed that the loss function landscape of deep neural networks is such that large-batch methods are attracted to regions with sharp minimizers and that, unlike small-batch methods, are unable to escape basins of attraction of these minimizers.

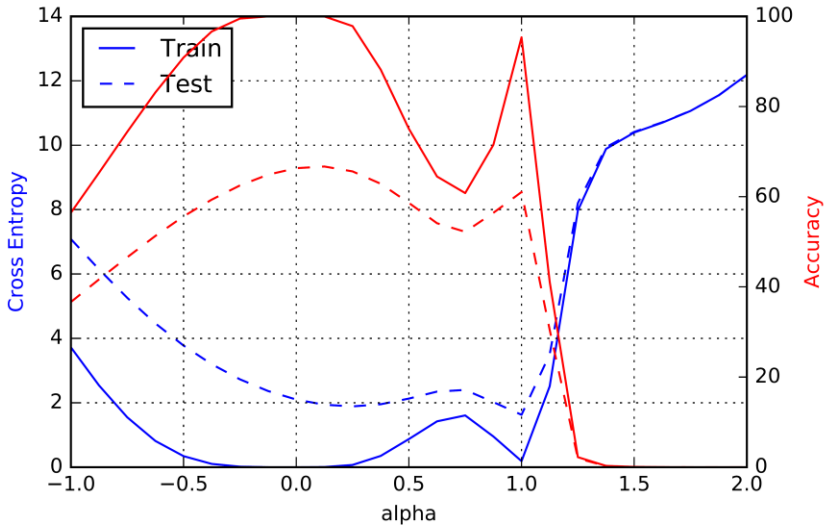


Sharpness Visualized in 1-D (Goodfellow et al., 2014b)

$$f(\alpha x_\ell^\star + (1-\alpha)x_s^\star) \qquad \alpha \in [-1, 2]$$



(a) F_1



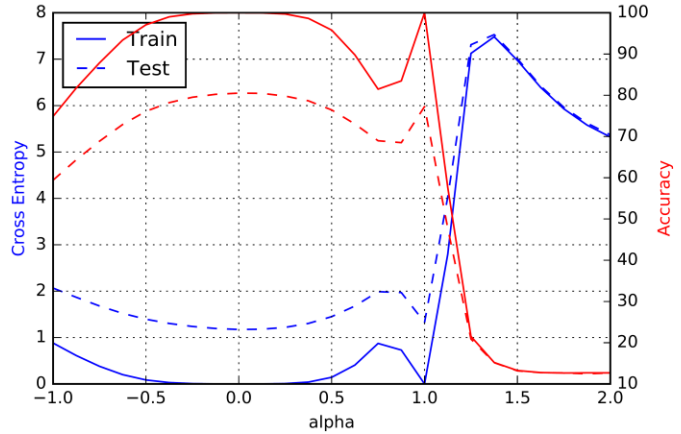
(b) F_2

	Training Accuracy		Testing Accuracy	
Name	SB	LB	SB	LB
F_1	$99.66\% \pm 0.05\%$	$99.92\% \pm 0.01\%$	$98.03\% \pm 0.07\%$	$97.81\% \pm 0.07\%$

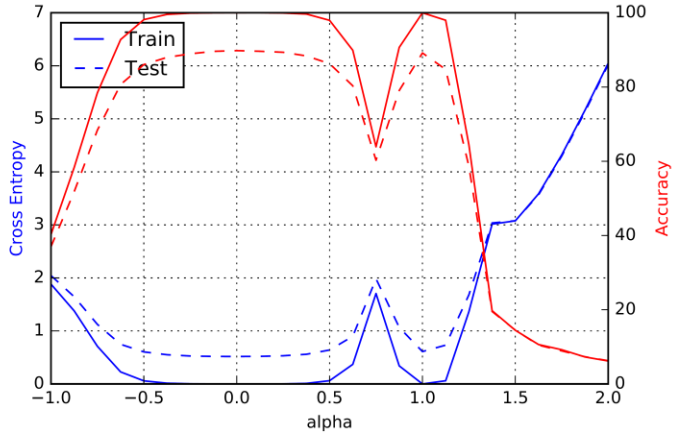
Sharpness Visualized in 1-D (Goodfellow et al., 2014b)

$$f(\alpha x_\ell^\star + (1-\alpha)x_s^\star)$$

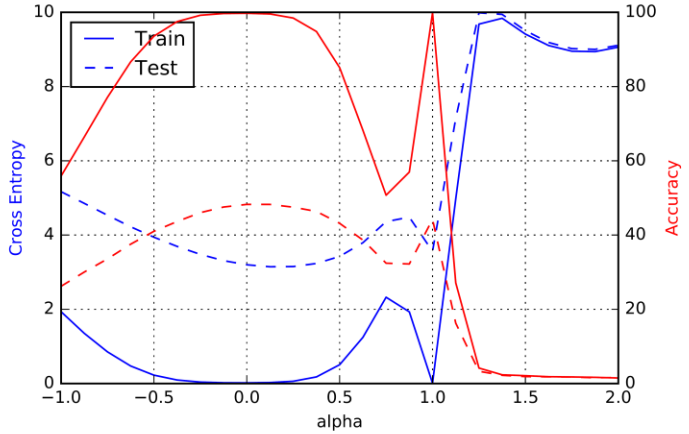
$\alpha \in [-1, 2]$



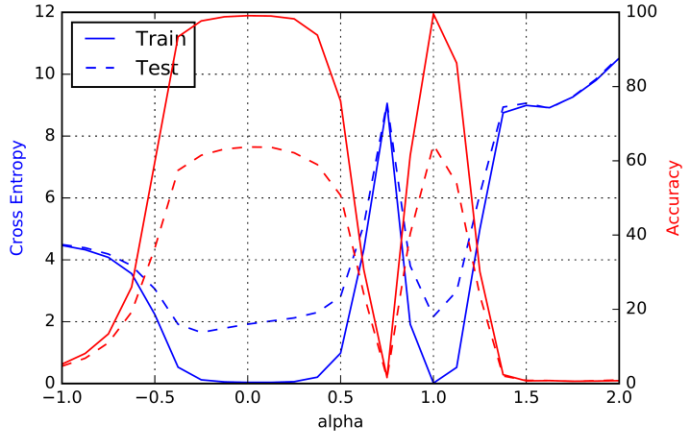
(c) C_1



(d) C_2



(e) C_3

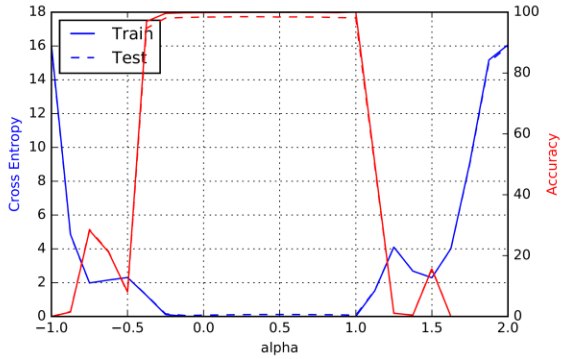


(f) C_4

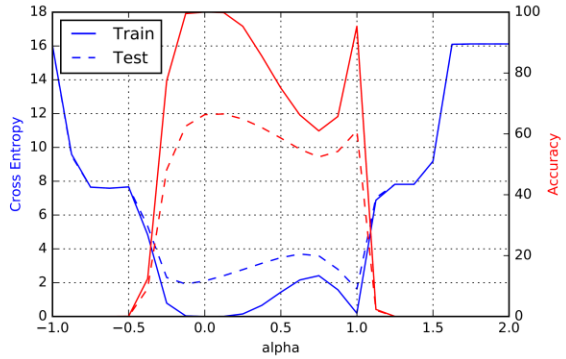
Curvilinear path

$$f(\sin(\tfrac{\alpha\pi}{2})x^\star_\ell + \cos(\tfrac{\alpha\pi}{2})x^\star_s)$$

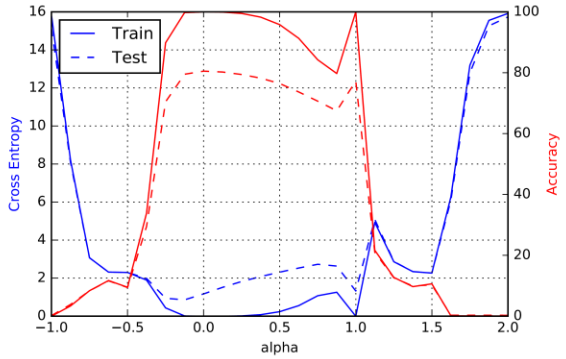
$\alpha \in [-1, 2]$



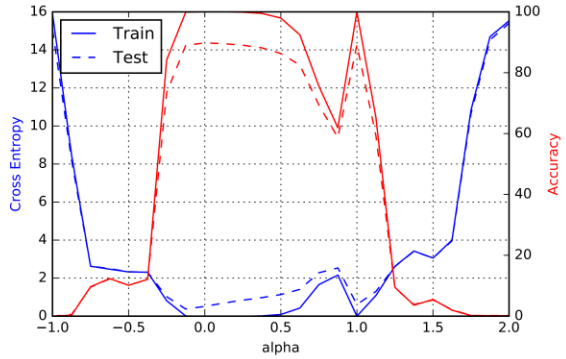
(a) F_1



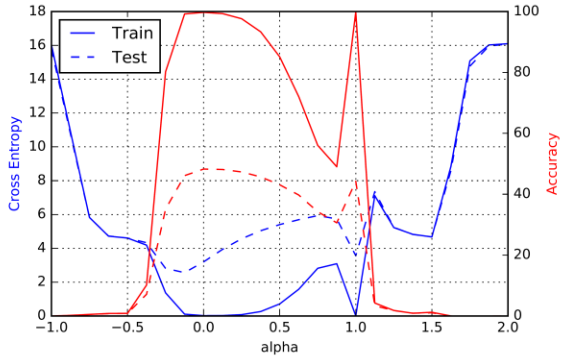
(b) F_2



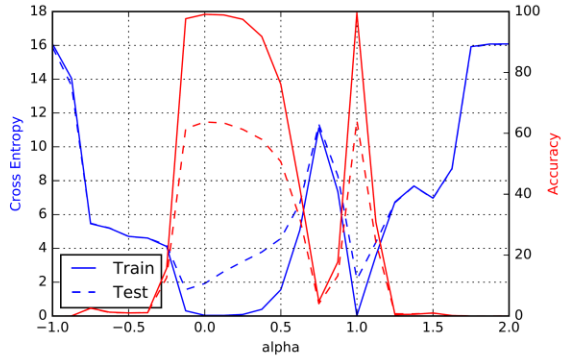
(c) C_1



(d) C_2



(e) C_3



(f) C_4

Sharpness Metric: Sensitivity measure

- Exploring a **small neighborhood** of a solution and computing the **largest value** that the function f can attain in that neighborhood.

Specifically, let $\mathcal{C}_\epsilon$ denote a box around the solution over which the maximization of f is performed, and let $A \in \mathbb{R}^{n\times p}$ be the matrix defined above. In order to ensure invariance of sharpness to problem dimension and sparsity, we define the constraint set $\mathcal{C}_\epsilon$ as:

$$\mathcal{C}_\epsilon = \{z \in \mathbb{R}^p : -\epsilon(|(A^+x)_i| + 1) \leq z_i \leq \epsilon(|(A^+x)_i| + 1) \quad \forall i \in \{1, 2, \cdots, p\}\}, \tag{3}$$

where A^+ denotes the pseudo-inverse of A. Thus ϵ controls the size of the box. We can now define our measure of sharpness (or sensitivity).

Metric 2.1. *Given $x \in \mathbb{R}^n$, $\epsilon > 0$ and $A \in \mathbb{R}^{n\times p}$, we define the $(\mathcal{C}_\epsilon, A)$-sharpness of f at x as:*

$$\phi_{x,f}(\epsilon, A) := \frac{(\max_{y\in\mathcal{C}_\epsilon} f(x + Ay)) - f(x)}{1 + f(x)} \times 100. \tag{4}$$

Sharpness Metric: Sensitivity measure

$$\phi_{x,f}(\epsilon, A) := \frac{(\max_{y \in \mathcal{C}_\epsilon} f(x + Ay)) - f(x)}{1 + f(x)} \times 100$$

- 2 scenarios:
  - Maximization over the entire space

    $A = I_n$
  - Random manifold:

    $A_{n \times p}$ matrix, randomly generated

    p: dimension of manifold (here p=100)
- $\epsilon = 1e-3$ and $\epsilon = 5e-5$

Table 3: Sharpness of Minima in Full Space; ϵ is defined in (3).

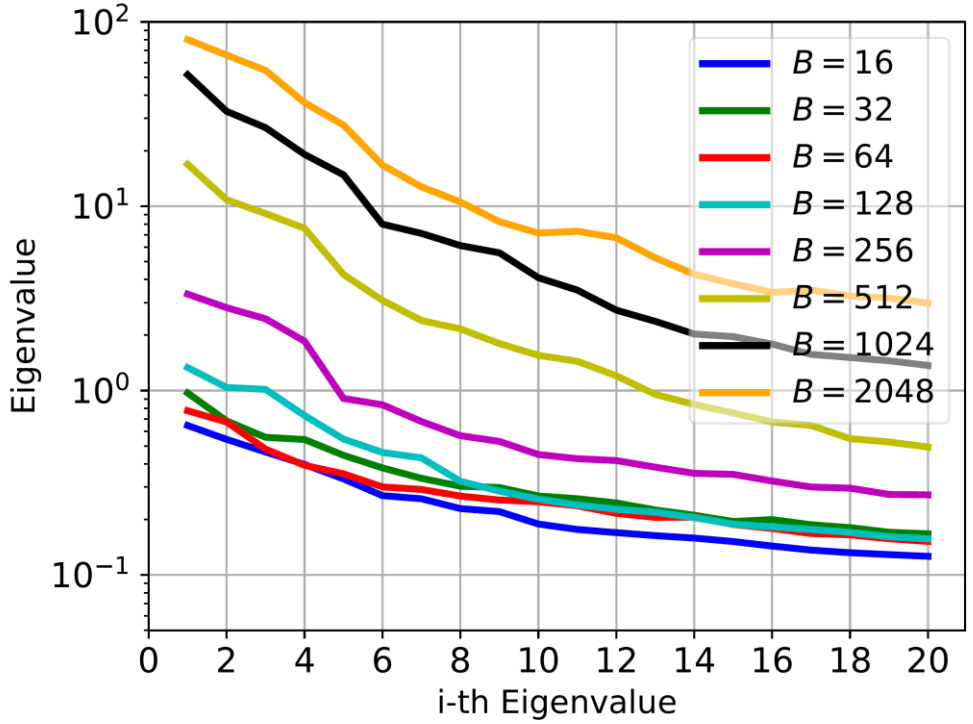
	$\epsilon = 10^{-3}$		$\epsilon = 5 \cdot 10^{-4}$	
	SB	LB	SB	LB
F_1	1.23 ± 0.83	205.14 ± 69.52	0.61 ± 0.27	42.90 ± 17.14
F_2	1.39 ± 0.02	310.64 ± 38.46	0.90 ± 0.05	93.15 ± 6.81
C_1	28.58 ± 3.13	707.23 ± 43.04	7.08 ± 0.88	227.31 ± 23.23
C_2	8.68 ± 1.32	925.32 ± 38.29	2.07 ± 0.86	175.31 ± 18.28
C_3	29.85 ± 5.98	258.75 ± 8.96	8.56 ± 0.99	105.11 ± 13.22
C_4	12.83 ± 3.84	421.84 ± 36.97	4.07 ± 0.87	109.35 ± 16.57

Table 4: Sharpness of Minima in Random Subspaces of Dimension 100

	$\epsilon = 10^{-3}$		$\epsilon = 5 \cdot 10^{-4}$	
	SB	LB	SB	LB
F_1	0.11 ± 0.00	9.22 ± 0.56	0.05 ± 0.00	9.17 ± 0.14
F_2	0.29 ± 0.02	23.63 ± 0.54	0.05 ± 0.00	6.28 ± 0.19
C_1	2.18 ± 0.23	137.25 ± 21.60	0.71 ± 0.15	29.50 ± 7.48
C_2	0.95 ± 0.34	25.09 ± 2.61	0.31 ± 0.08	5.82 ± 0.52
C_3	17.02 ± 2.20	236.03 ± 31.26	4.03 ± 1.45	86.96 ± 27.39
C_4	6.05 ± 1.13	72.99 ± 10.96	1.89 ± 0.33	19.85 ± 4.12

Hessian-based Analysis of Large Batch Training and Robustness to Adversaries (Yao et al. NeurIPS 2018)

- Directly computing the **spectrum of the true Hessian**, and show that large-batch gets trapped in areas with noticeably larger spectrum
- Models trained with large batch size are significantly more prone to **adversarial attacks**



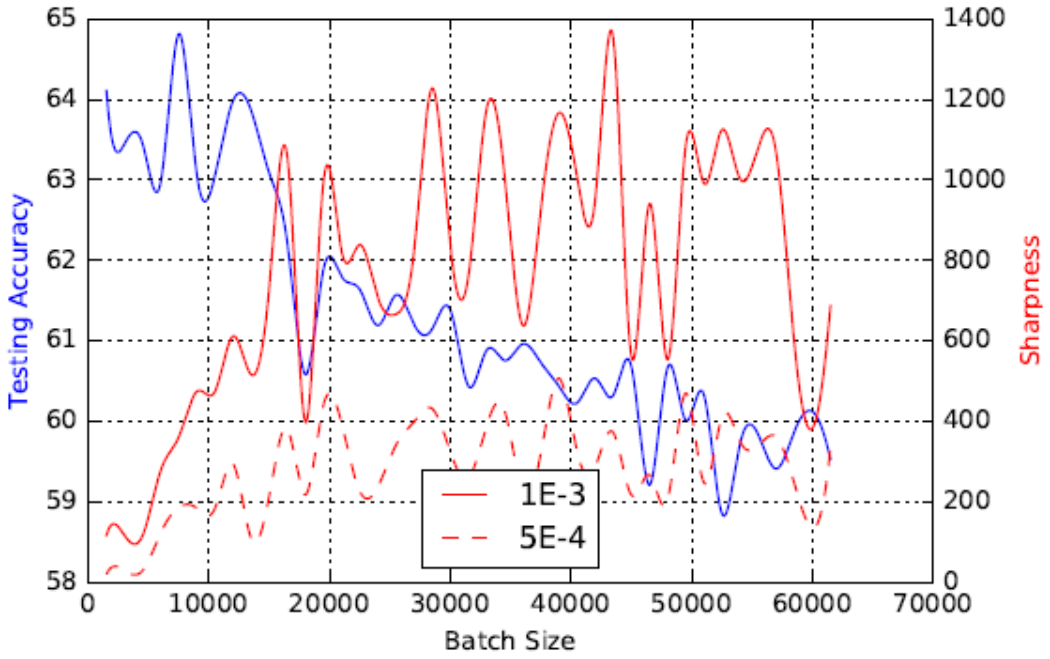
	Batch	Acc.	λ_1^θ
C1 Cifar-10	16	100 (77.68)	0.64 (32.78)
	32	100 (76.77)	0.97 (45.28)
	64	100 (77.32)	0.77 (48.06)
	128	100 (78.84)	1.33 (137.5)
	256	100 (78.54)	3.34 (338.3)
	512	100 (79.25)	16.88 (885.6)
	1024	100 (78.50)	51.67 (2372)
	2048	100 (77.31)	80.18 (3769)

Test result is given in parentheses

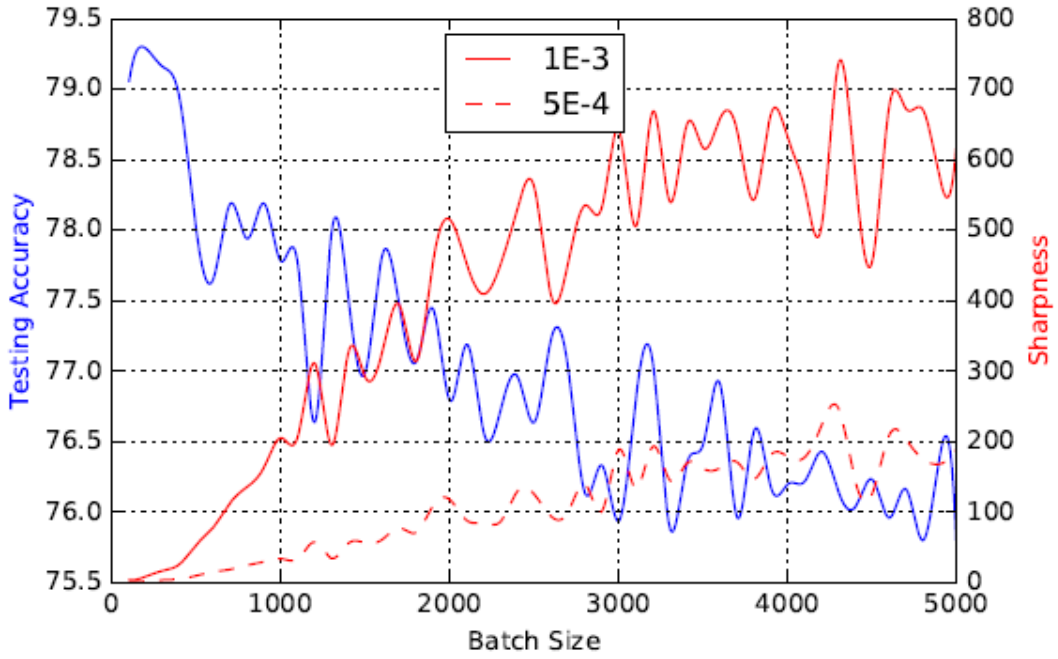
Sharpness of Minima: Sensitivity measure

- Sharp minimizers DO NOT resemble a **cone**
  - Function does not increase rapidly along all directions.
  - It rises steeply only along a **small dimensional subspace** (e.g. 5% of the whole space)
  - On most other directions, the function is relatively **flat**

Batch-size Vs. Sharpness and Accuracy



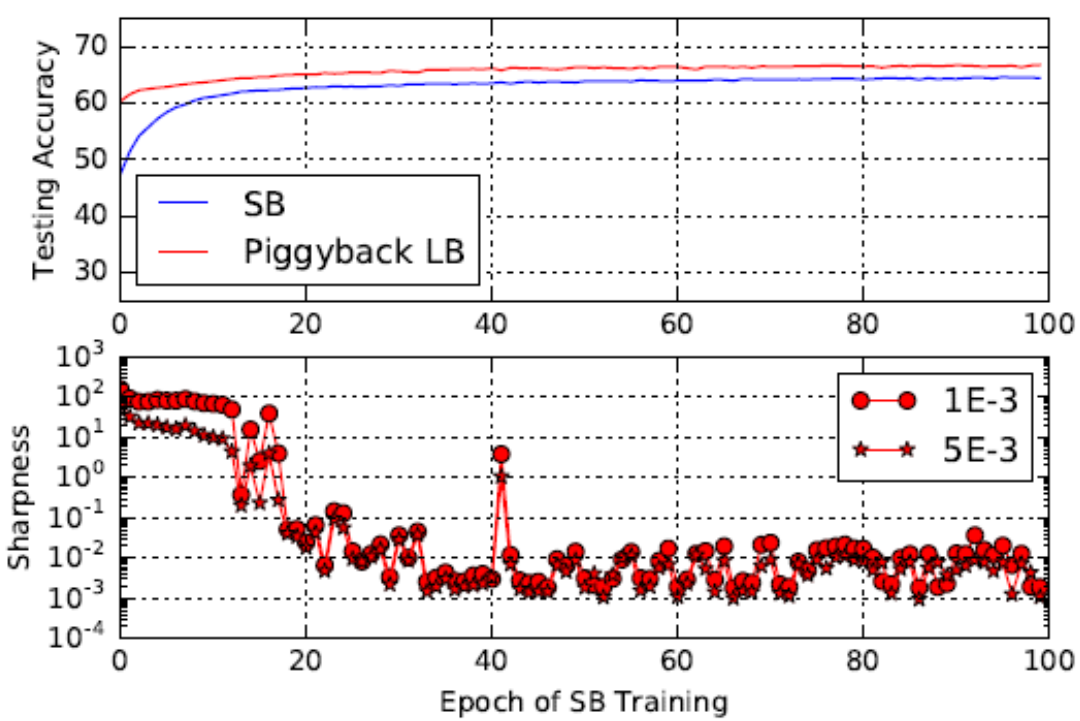
(a) F_2



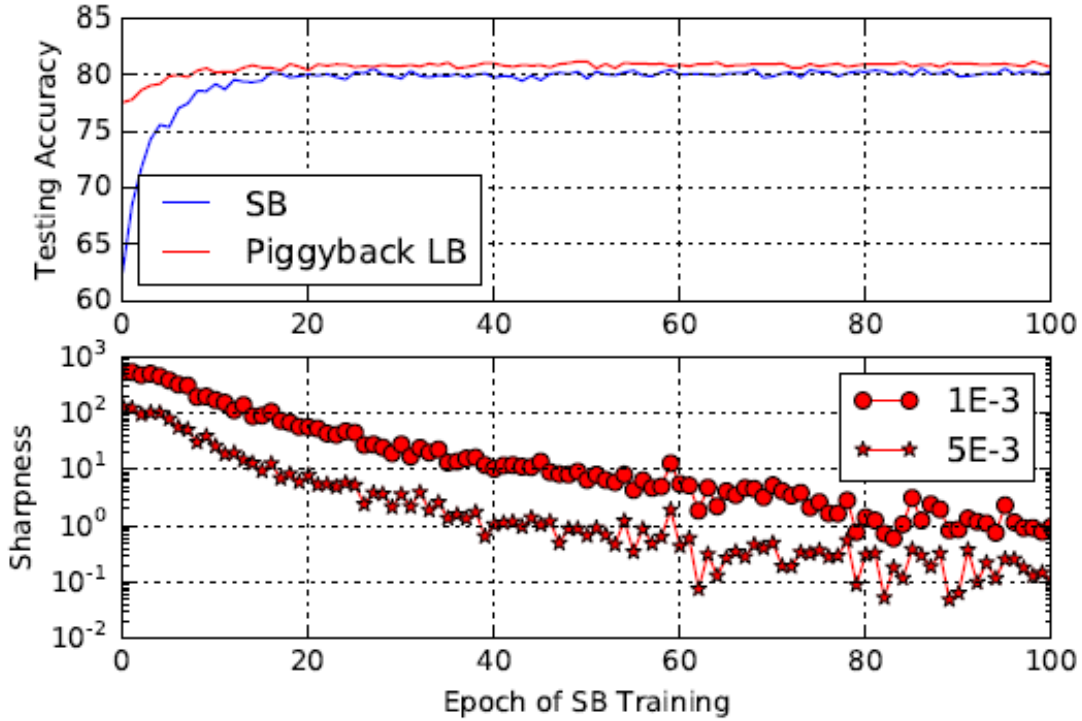
(b) C_1

Warm-started Large-batch

Warm-start the training with 0 to 100 epochs of small-batch
Continue with large-batch training until convergence.



(a) F_2



(b) C_1

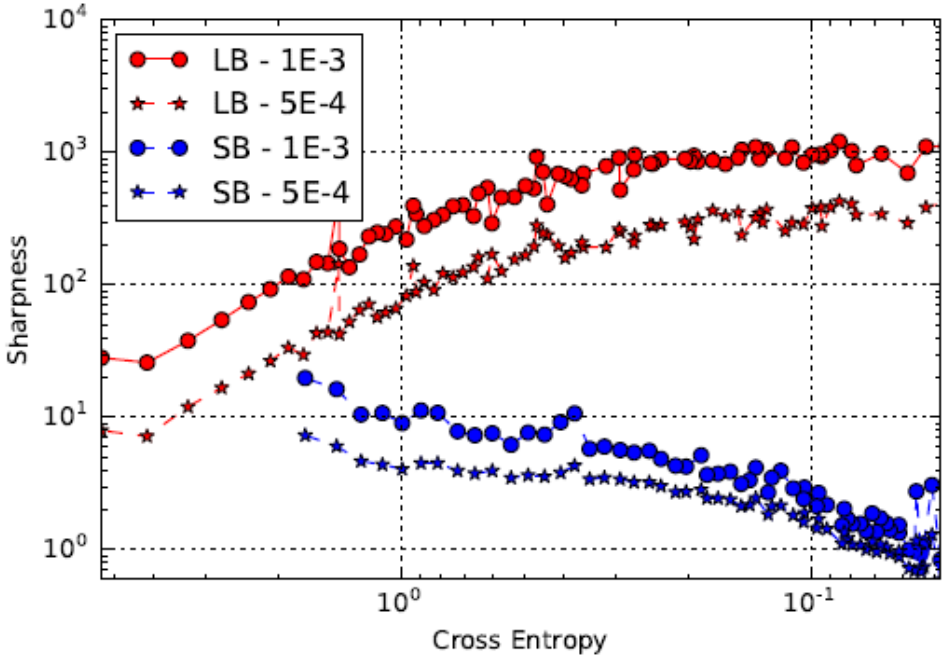
Note: Dynamic sampling where the **batch-size is increased gradually** (Byrd et al., 2012; Friedlander & Schmidt, 2012)

Distance of the Converged Optimizer to the Initial Point

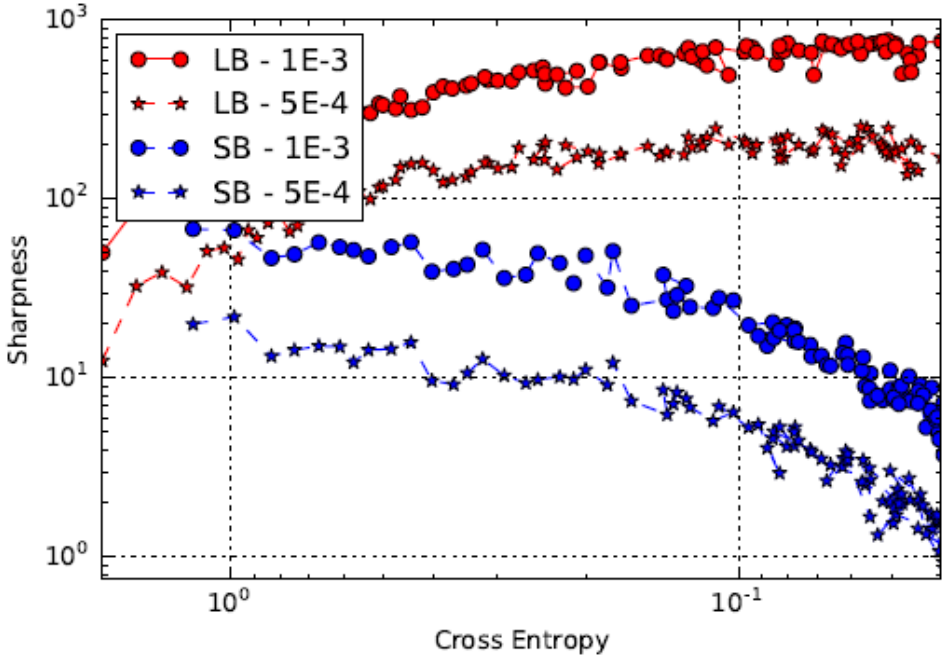
- It has been speculated that LB methods tend to be attracted to minimizers close to the starting point x0, whereas SB methods move farther away.
- Observed that the ratio of $\|x_s^\star - x_0\|_2$ and $\|x_\ell^\star - x_0\|_2$ was in the range of 3–10.

Sharpness Vs. Loss

- Near the initial point, SB and LB method yield similar values of sharpness.
- As the loss function reduces,
  - the sharpness of LB **increases**,
  - the sharpness of SB stays relatively constant initially and then **reduces**



(a) F_2



(b) C_1

(Appendix)

So, what is the solution?

(of course, except for reducing the batch-size!)

(Appendix)

Mitigating the Generalization Gap
Data Augmentation

- Without Augmentation

Name	Testing Accuracy SB	Testing Accuracy LB	Sharpness (LB method) $\epsilon = 10^{-3}$	Sharpness (LB method) $\epsilon = 5 \cdot 10^{-4}$
C_1	$80.04\% \pm 0.12\%$	$77.26\% \pm 0.42\%$	707.23 ± 43.04	227.31 ± 23.23
C_2	$89.24\% \pm 0.12\%$	$87.26\% \pm 0.07\%$	925.32 ± 38.29	175.31 ± 18.28
C_3	$49.58\% \pm 0.39\%$	$46.45\% \pm 0.43\%$	258.75 ± 8.96	105.11 ± 13.22
C_4	$63.08\% \pm 0.5\%$	$57.81\% \pm 0.17\%$	421.84 ± 36.97	109.35 ± 16.57

- With Augmentation
  - horizontal reflections,
  - random rotations up to 10
  - random translation of up to 0.2 times the size of the image

	Testing Accuracy Baseline (SB)	Testing Accuracy Augmented LB	Sharpness (LB method) $\epsilon = 10^{-3}$	Sharpness (LB method) $\epsilon = 5 \cdot 10^{-4}$
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

(Appendix)

Mitigating the Generalization Gap
Conservative Training (Li et al., 2014)

- Without Conservative training

Name	Testing Accuracy SB	Testing Accuracy LB	Sharpness (LB method) $\epsilon = 10^{-3}$	Sharpness (LB method) $\epsilon = 5 \cdot 10^{-4}$
F_1	$98.03\% \pm 0.07\%$	$97.81\% \pm 0.07\%$	205.14 ± 69.52	42.90 ± 17.14
F_2	$64.02\% \pm 0.2\%$	$59.45\% \pm 1.05\%$	310.64 ± 38.46	93.15 ± 6.81
C_1	$80.04\% \pm 0.12\%$	$77.26\% \pm 0.42\%$	707.23 ± 43.04	227.31 ± 23.23
C_2	$89.24\% \pm 0.12\%$	$87.26\% \pm 0.07\%$	925.32 ± 38.29	175.31 ± 18.28
C_3	$49.58\% \pm 0.39\%$	$46.45\% \pm 0.43\%$	258.75 ± 8.96	105.11 ± 13.22
C_4	$63.08\% \pm 0.5\%$	$57.81\% \pm 0.17\%$	421.84 ± 36.97	109.35 ± 16.57

- With Conservative training:
  - better utilize a batch before moving onto the next one.
  - Using 3 iterations of ADAM
  - $\lambda = 1e - 3$
  - Solve this proximal sub-problem

$$x_{k+1} = \arg\min_x \frac{1}{|B_k|} \sum_{i \in B_k} f_i(x) + \frac{\lambda}{2} \|x - x_k\|_2^2$$

	Testing Accuracy Baseline (SB)	Testing Accuracy Conservative LB	Sharpness (LB method) $\epsilon = 10^{-3}$	Sharpness (LB method) $\epsilon = 5 \cdot 10^{-4}$
F_1	$98.03\% \pm 0.07\%$	$98.12\% \pm 0.01\%$	232.25 ± 63.81	46.02 ± 12.58
F_2	$64.02\% \pm 0.2\%$	$61.94\% \pm 1.10\%$	928.40 ± 51.63	190.77 ± 25.33
C_1	$80.04\% \pm 0.12\%$	$78.41\% \pm 0.22\%$	520.34 ± 34.91	171.19 ± 15.13
C_2	$89.24\% \pm 0.05\%$	$88.495\% \pm 0.63\%$	632.01 ± 208.01	108.88 ± 47.36
C_3	$49.58\% \pm 0.39\%$	$45.98\% \pm 0.54\%$	337.92 ± 33.09	110.69 ± 3.88
C_4	$63.08\% \pm 0.10\%$	62.51 ± 0.67	354.94 ± 20.23	68.76 ± 16.29

(Appendix)

Idealized Performance Model

For LB to be faster than SB: $$I_\ell \frac{B_\ell}{P} < I_s \frac{B_s}{P f_s(P)}$$

- Let I_s and I_l : number of iterations required by SB and LB methods to converge
- Let B_s and B_l : Batch sizes of SB and LB methods
- P: Number of processors
- $f_s(P)$: relative parallel efficiency of the SB method
- $f_l(P)$: parallel efficiency of the LB method (assumed to be equal to 1.0)

Open Questions

a) Can one **prove** that large-batch (LB) methods typically converge to sharp minimizers of deep learning training functions?

b) What is the **relative density** of the two kinds of minima?

c) Can one **design neural network architectures** for various tasks that are suitable to the properties of LB methods?

d) Can the networks be **initialized** in a way that enables LB methods to succeed?

e) Is it possible, through algorithmic or regulatory means to **steer LB methods away** from sharp minimizers?

- How does very small batches affect generalization?
  - Response on Open Review: From our preliminary experiments, it seems that there is no significant benefit from reducing batch-sizes to a very small value (**8 and 16 are similar to 256**)

Main Takeaways

Based on this research's empirical observations,

- Large-batch (LB) methods
  - Lack the **explorative** properties of small-batch (SB) methods,
  - Tend to zoom-in on the minimizer **closest to the initial point**
  - converge to **sharp minimizers** with differing generalization properties
- The **generalization gap is correlated with the sharpness** of the minimizers
- Data augmentation and Conservative training are ineffective in reducing sharpness

Thank you