Large-Batch Training and Generalization Gap

Nitish Shirish Keskar
Salesforce Research

Slides: keskarnitish.github.io

Deep Learning & SGD

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

Deep Learning & SGD

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

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

Deep Learning & SGD

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

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

- Large batch sizes ⇒ more concurrency ⇒ easier (data) parallelization ⇒ reduced time-to-solution

Deep Learning & SGD

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

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

- Large batch sizes ⇒ more concurrency ⇒ easier (data) parallelization ⇒ reduced time-to-solution

- Other ways to parallelize, not the focus of this talk.

Concern With Using Large Batches

Concern With Using Large Batches

Standard Architecture + Standard Training + LB = Great Training, Deficient Testing Performance

Concern With Using Large Batches

Standard Architecture + Standard Training + LB = Great Training, Deficient Testing Performance

Standard Architecture + Standard Training + SB = Great Training, Great Testing Performance

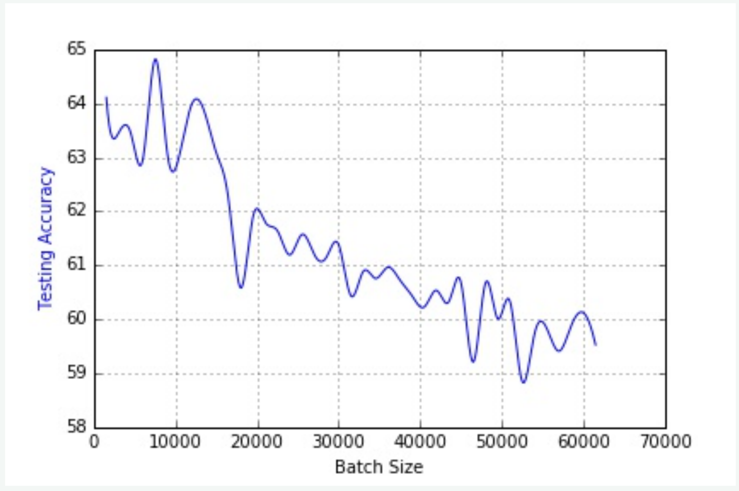
Concern With Using Large Batches

Standard Architecture + Standard Training + LB = Great Training, Deficient Testing Performance

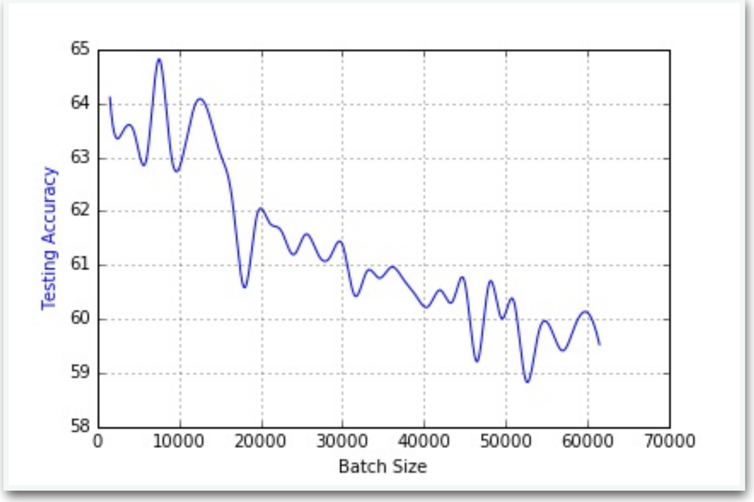
Standard Architecture + Standard Training + SB = Great Training, Great Testing Performance

Changing the training regime and/or architecture can alleviate this problem, subject of the next few talks.

Name	Training Accuracy SB	LB	Testing Accuracy 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	$99.10\% \pm 1.23\%$	$99.57\% \pm 1.84\%$	$63.08\% \pm 0.5\%$	$57.81\% \pm 0.17\%$



	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	$99.10\% \pm 1.23\%$	$99.57\% \pm 1.84\%$	$63.08\% \pm 0.5\%$	$57.81\% \pm 0.17\%$

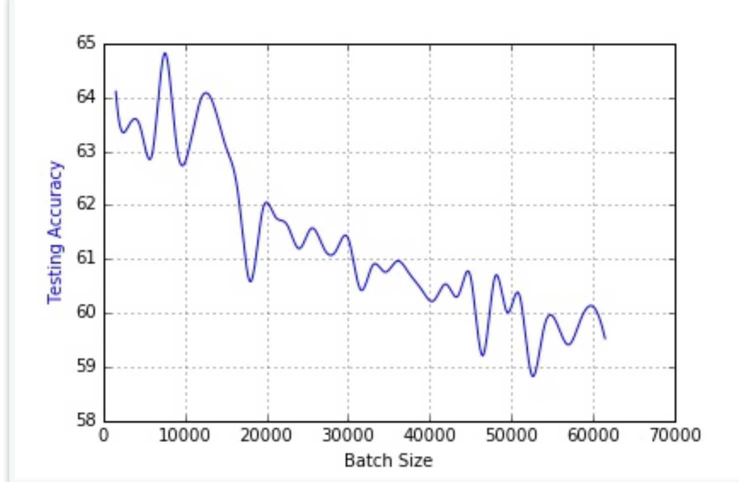


Network	LB size	Dataset	SB	LB[6]	+LR[6]
Alexnet	4096	ImageNet	57.10%	41.23%	53.25%
Alexnet	8192	ImageNet	57.10%	41.23%	53.25%

Network	Dataset	SB	LB	+LR
F1 (Keskar et al., 2017)	MNIST	98.27%	97.05%	97.55%
C1 (Keskar et al., 2017)	Cifar10	87.80%	83.95%	86.15%
Resnet44 (He et al., 2016)	Cifar10	92.83%	86.10%	89.30%
VGG (Simonyan, 2014)	Cifar10	92.30%	84.1%	88.6%
C3 (Keskar et al., 2017)	Cifar100	61.25%	51.50%	57.38%
WResnet16-4 (Zagoruyko, 2016)	Cifar100	73.70%	68.15%	69.05%

Hoffer et. al. (2017)

	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	$99.10\% \pm 1.23\%$	$99.57\% \pm 1.84\%$	$63.08\% \pm 0.5\%$	$57.81\% \pm 0.17\%$



Network	LB size	Dataset	SB	LB[6]	+LR[6]
Alexnet	4096	ImageNet	57.10%	41.23%	53.25%
Alexnet	8192	ImageNet	57.10%	41.23%	53.25%

Network	Dataset	SB	LB	+LR
F1 (Keskar et al., 2017)	MNIST	98.27%	97.05%	97.55%
C1 (Keskar et al., 2017)	Cifar10	87.80%	83.95%	86.15%
Resnet44 (He et al., 2016)	Cifar10	92.83%	86.10%	89.30%
VGG (Simonyan, 2014)	Cifar10	92.30%	84.1%	88.6%
C3 (Keskar et al., 2017)	Cifar100	61.25%	51.50%	57.38%
WResnet16-4 (Zagoruyko, 2016)	Cifar100	73.70%	68.15%	69.05%

Hoffer et. al. (2017)

	k	n	kn	η	top-1 error (%)
baseline (single server)	8	32	256	0.1	23.60 ±0.12
no warmup, Figure 2a	256	32	8k	3.2	24.84 ±0.37

Goyal et. al. (2017)

The “Why” Hypothesis
Flavors of Minima

The “Why” Hypothesis
Flavors of Minima

- LB training ⇒ convergence to *sharp minima* of the loss.

The “Why” Hypothesis
Flavors of Minima

- LB training ⇒ convergence to *sharp minima* of the loss.
- Sharp minima *correlate* with poor generalization.

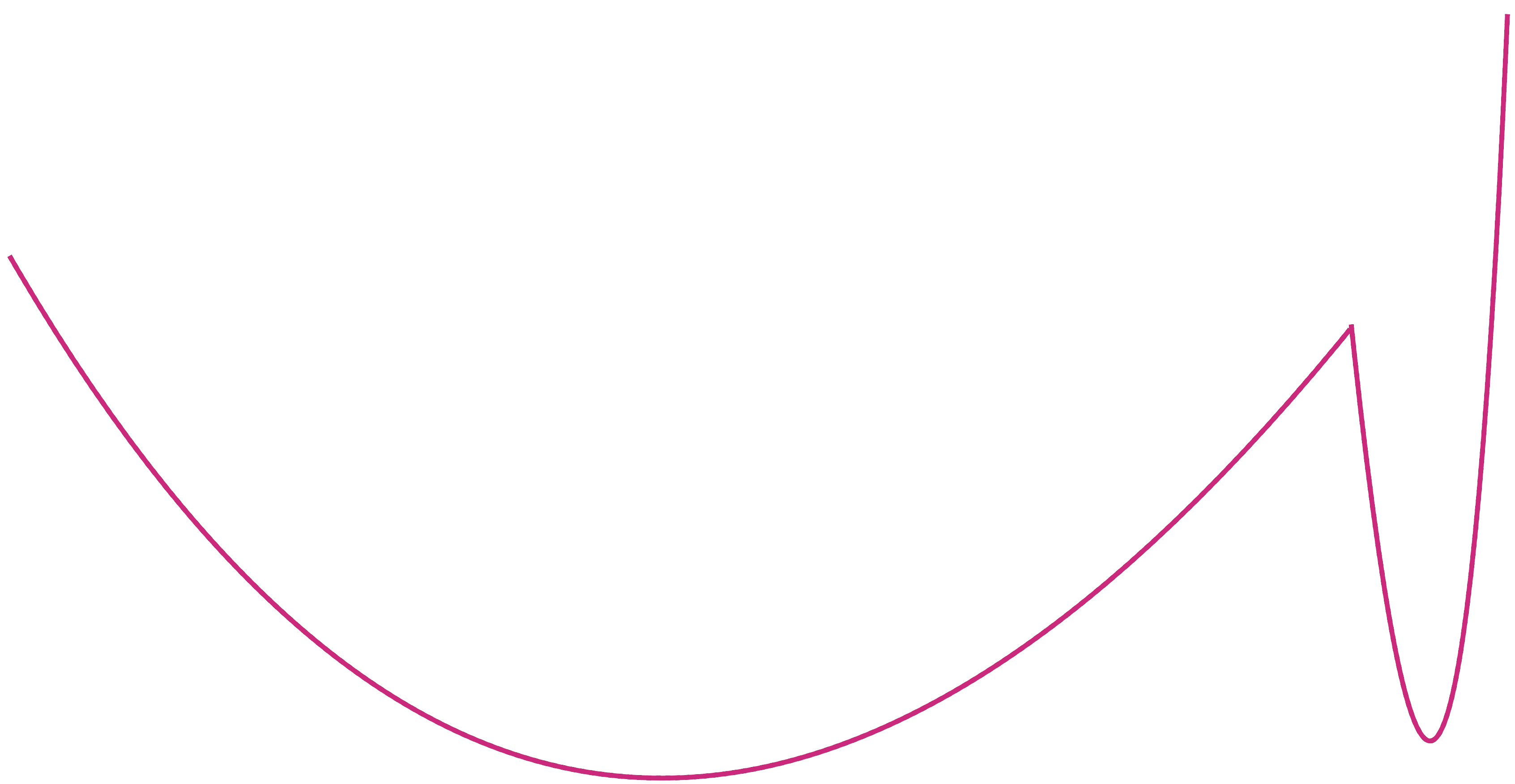
The “Why” Hypothesis
Flavors of Minima

- LB training $\Rightarrow$ convergence to *sharp minima* of the loss.
- Sharp minima *correlate* with poor generalization.
- SB training avoids such minima due to update noise.

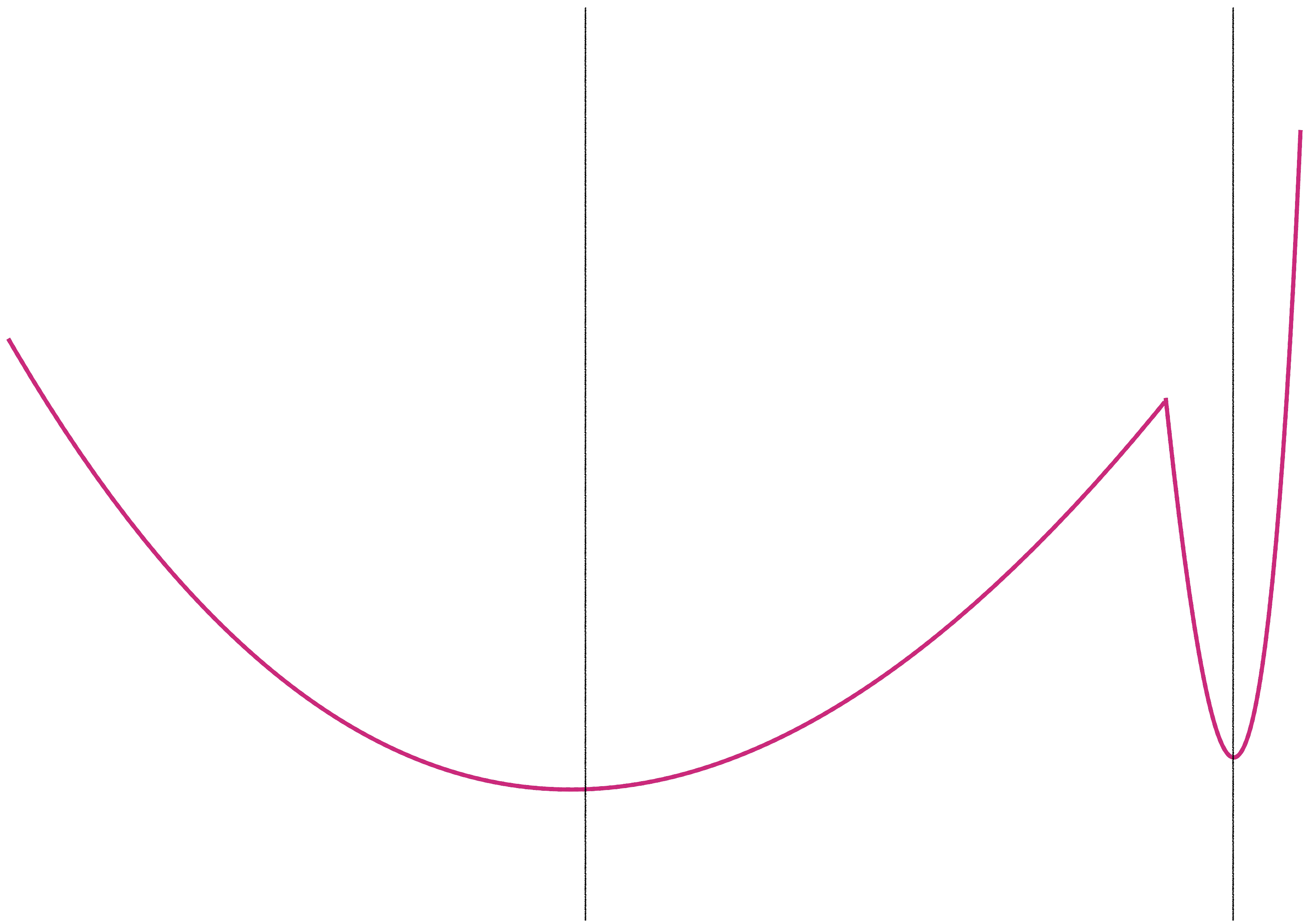
The “Why” Hypothesis Flavors of Minima

- LB training ⇒ convergence to *sharp minima* of the loss.
- Sharp minima *correlate* with poor generalization.
- SB training avoids such minima due to update noise.
- Evidence in support: parametric plots and sharpness metric.

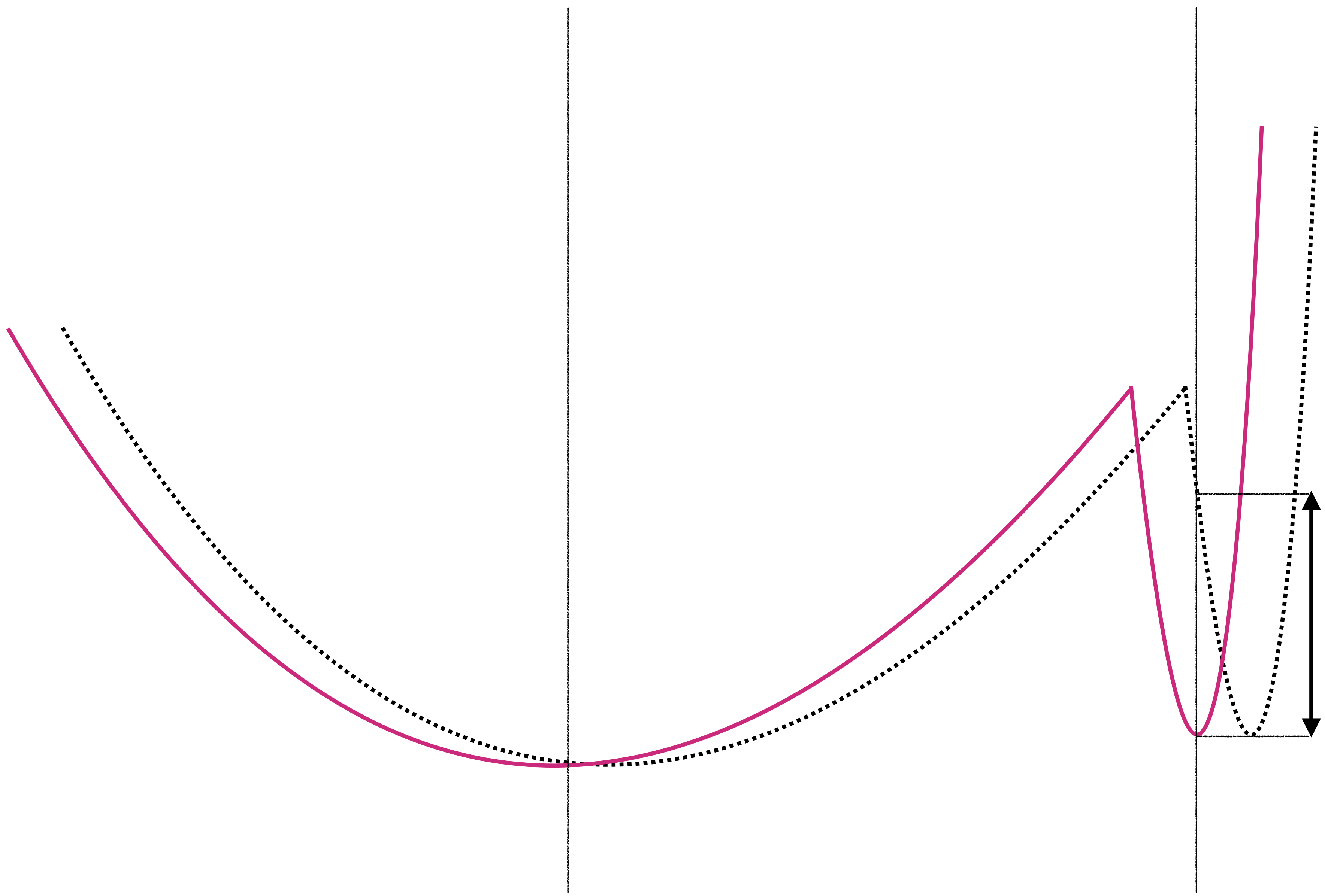
What's Wrong With Sharp Minima? A Simplistic Explanation



What's Wrong With Sharp Minima? A Simplistic Explanation

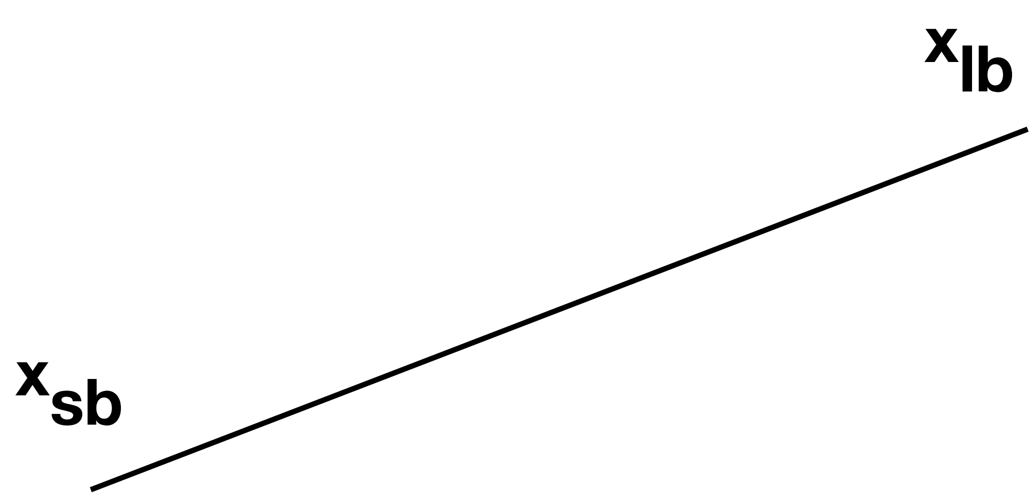


What's Wrong With Sharp Minima?
A Simplistic Explanation

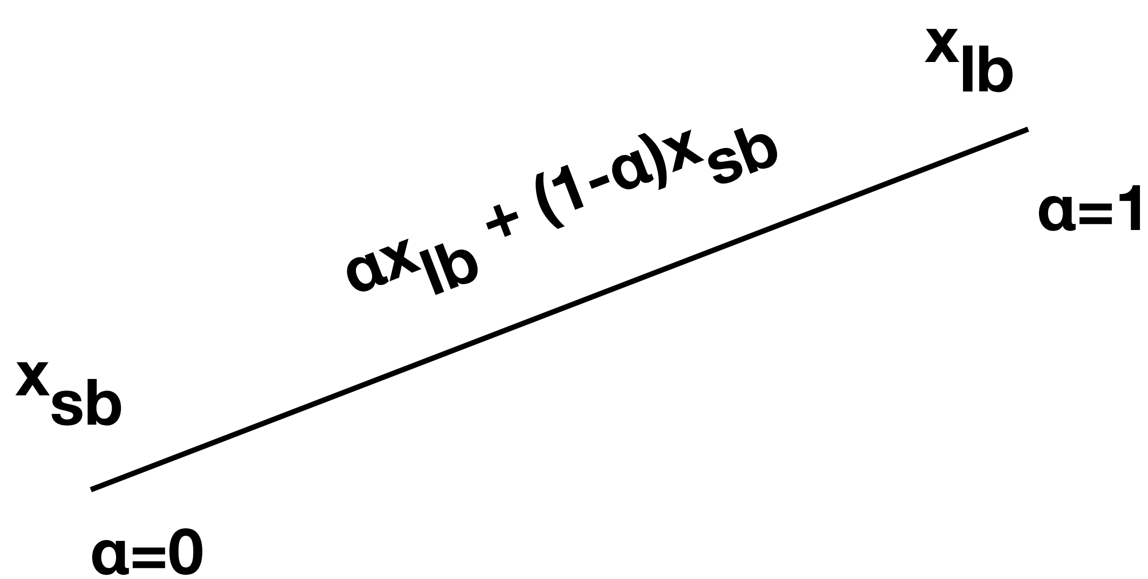


Parametric Plots

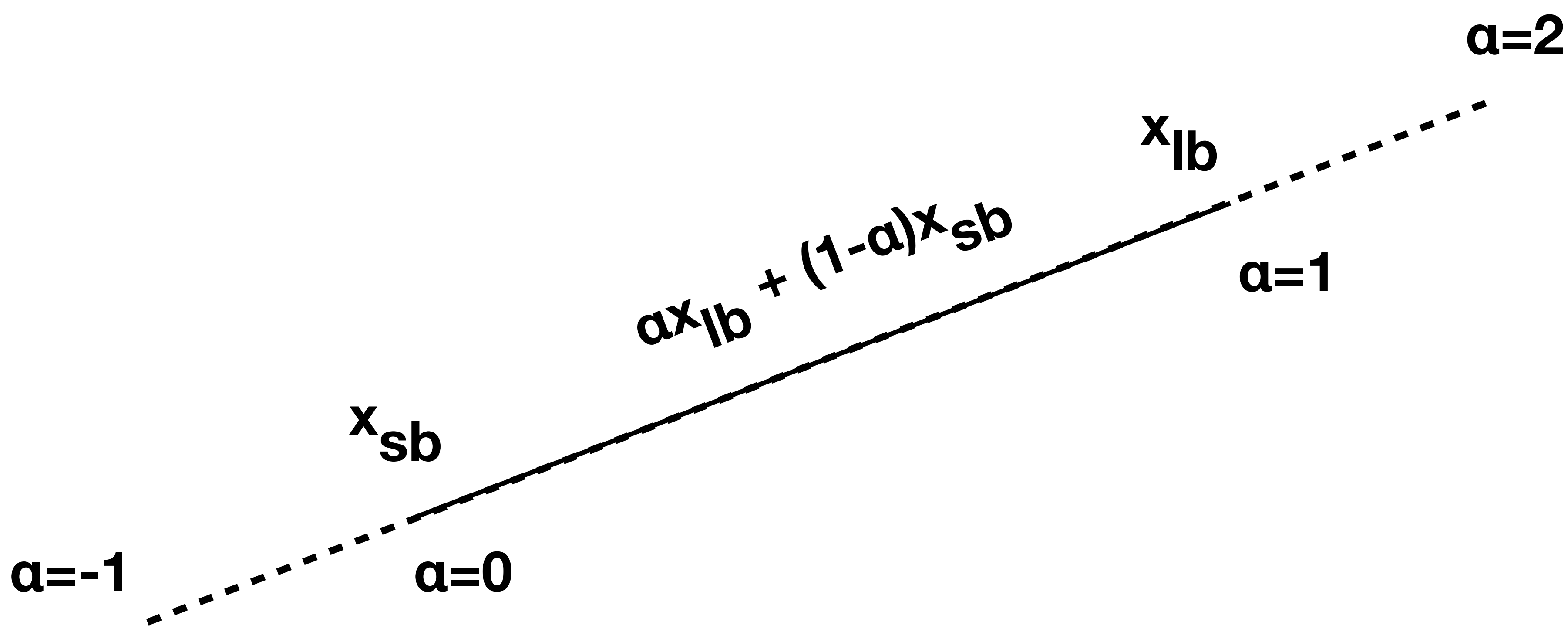
Parametric Plots



Parametric Plots

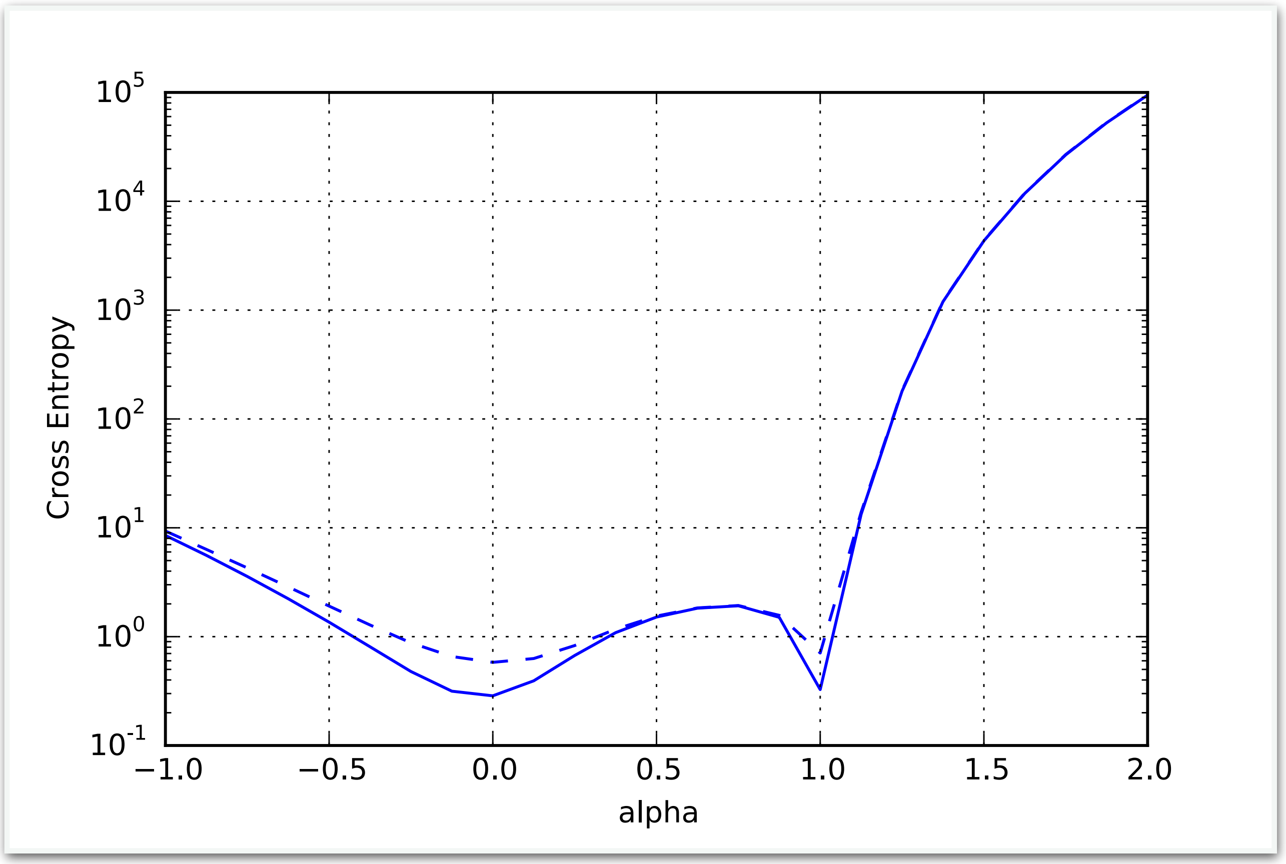


Parametric Plots



Deep ConvNet on CIFAR10

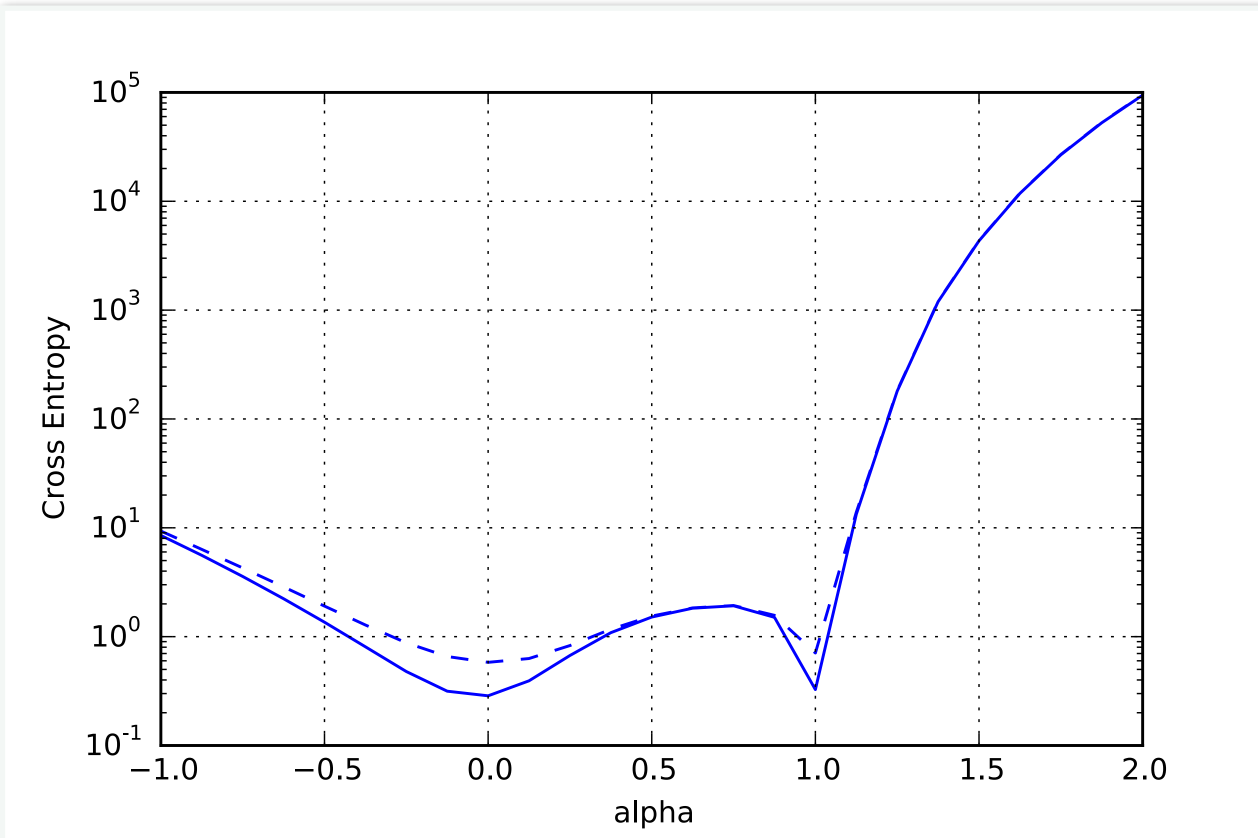
Deep ConvNet on CIFAR10

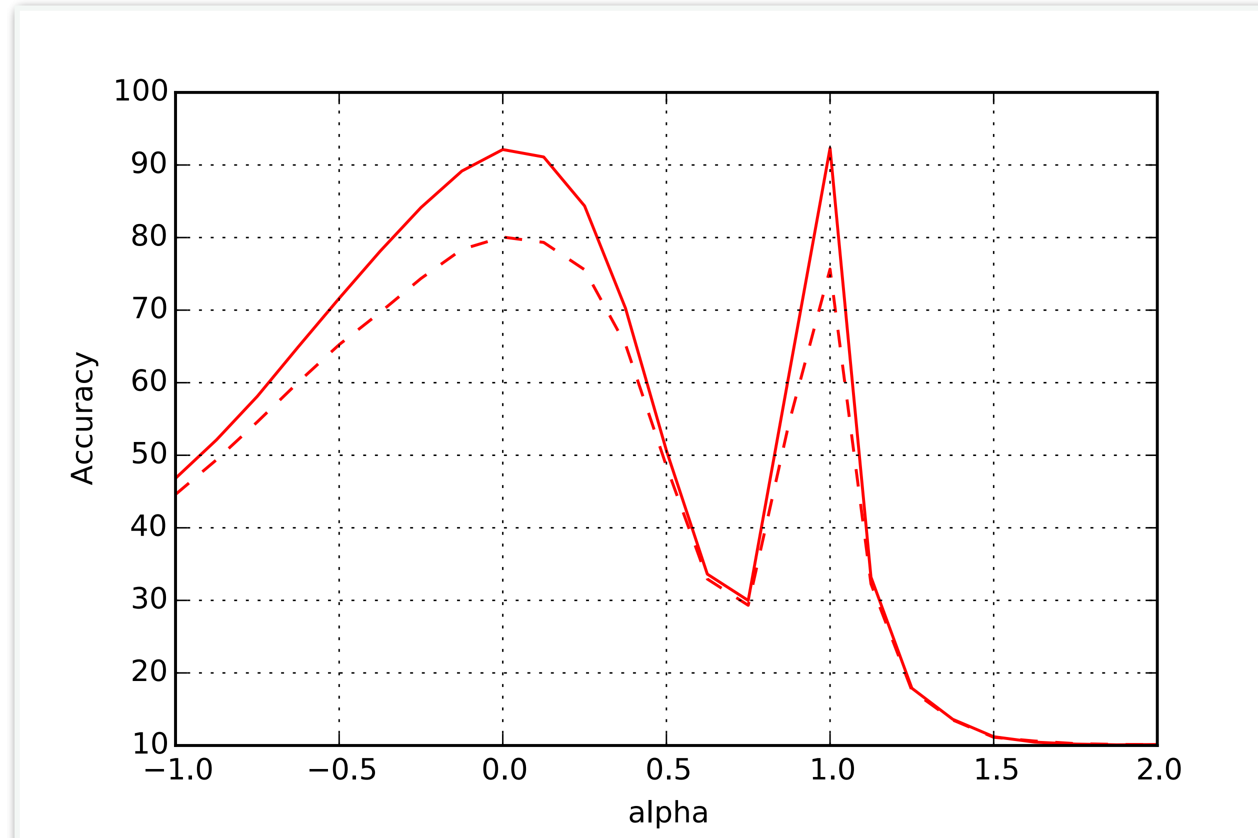


SB Solution

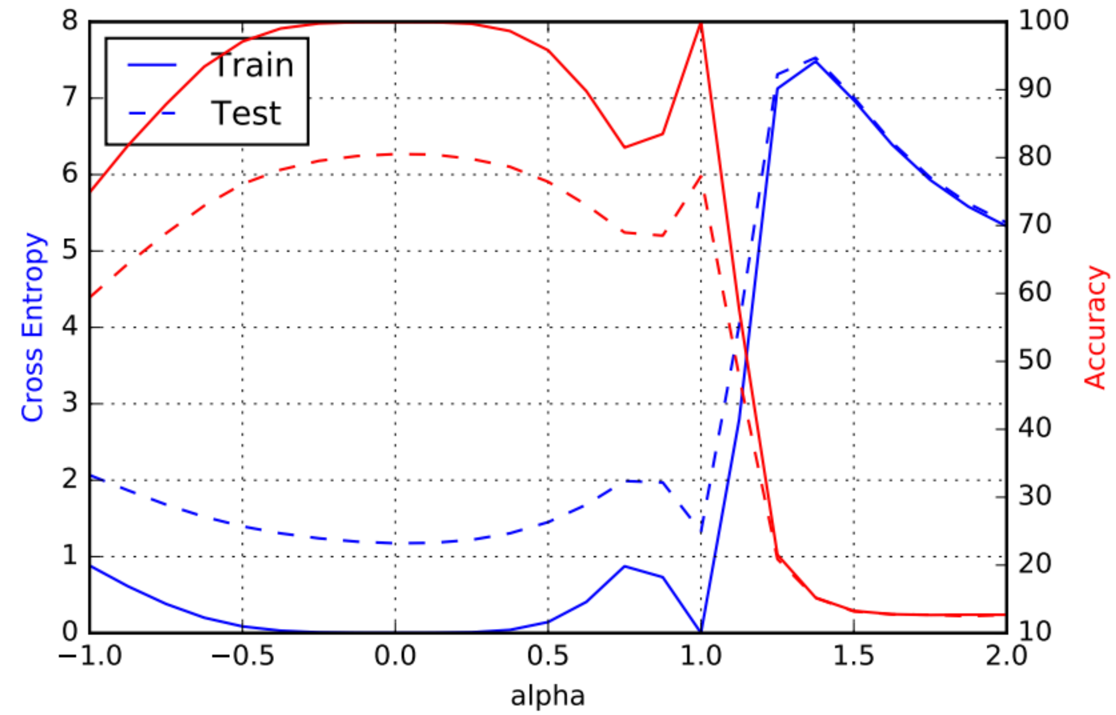
LB Solution

Deep ConvNet on CIFAR10

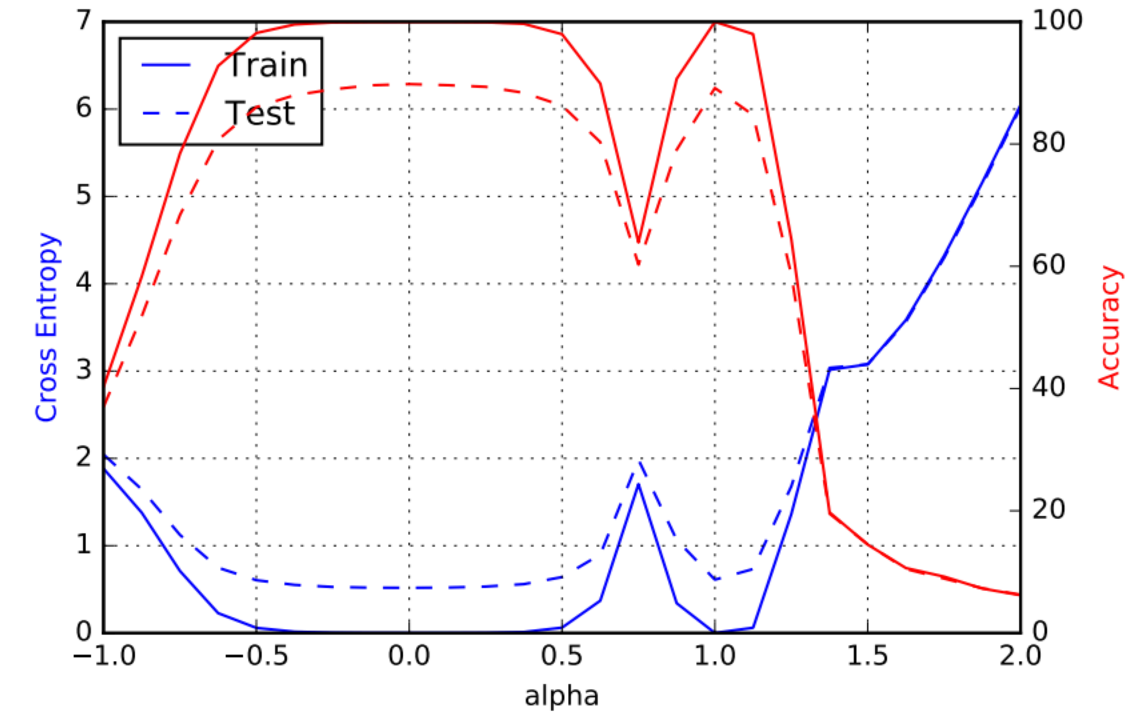




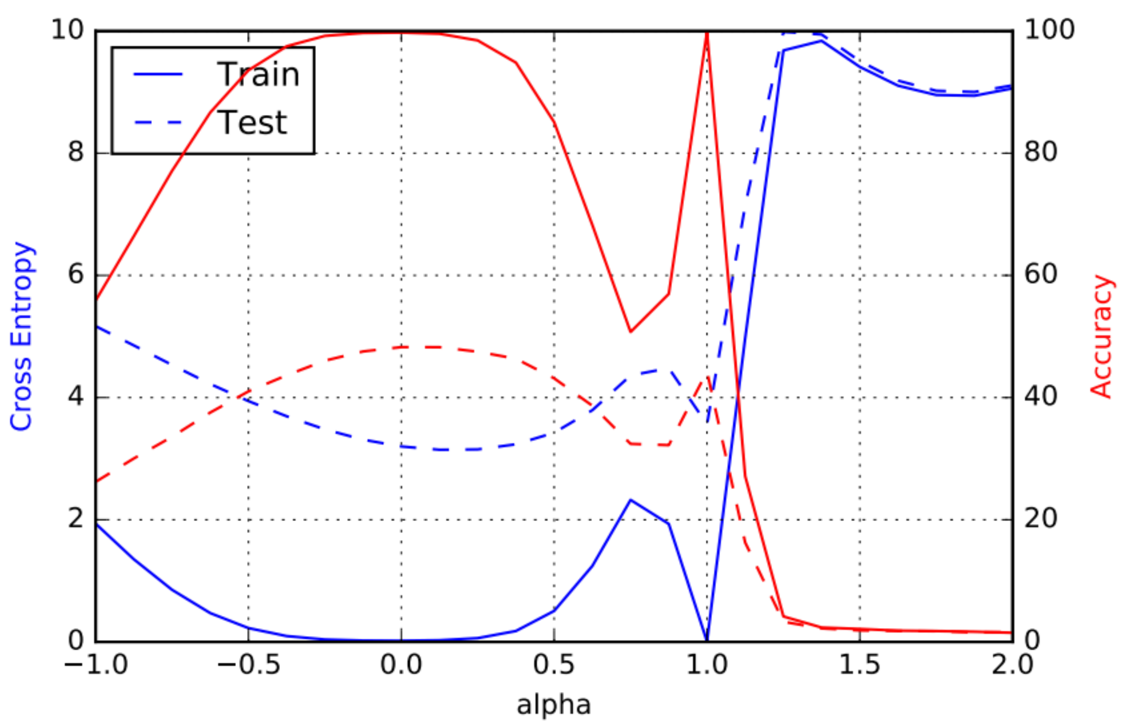
SB Solution LB Solution SB Solution LB Solution



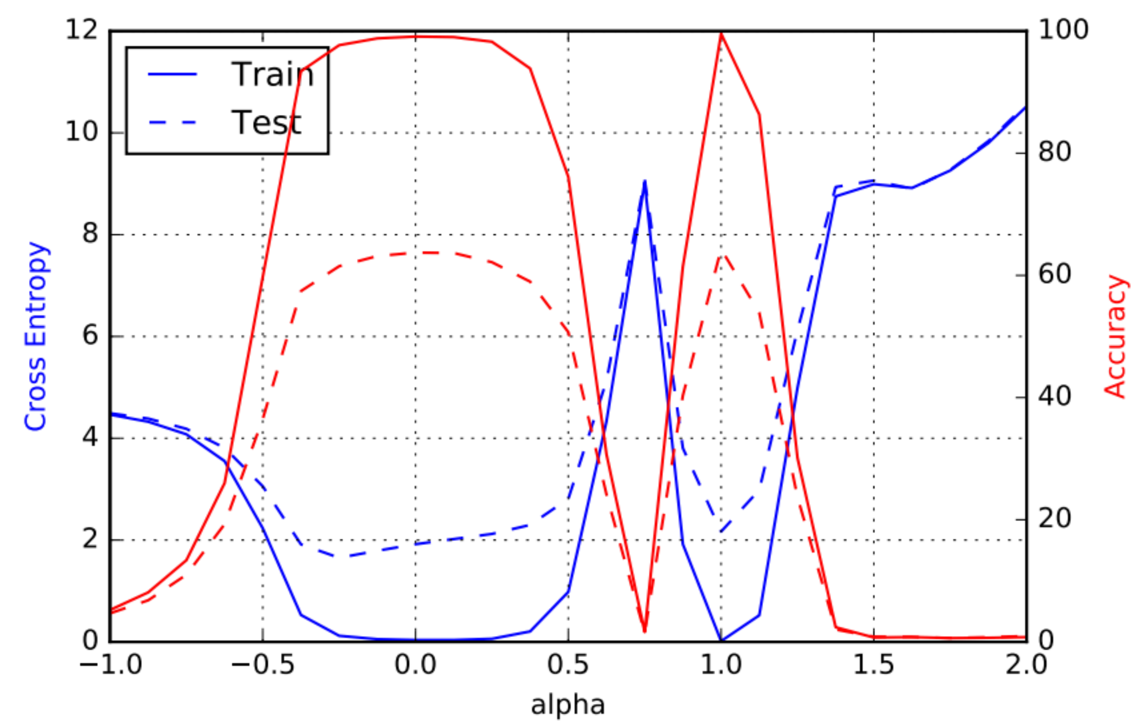
(c) C_1



(d) C_2



(e) C_3



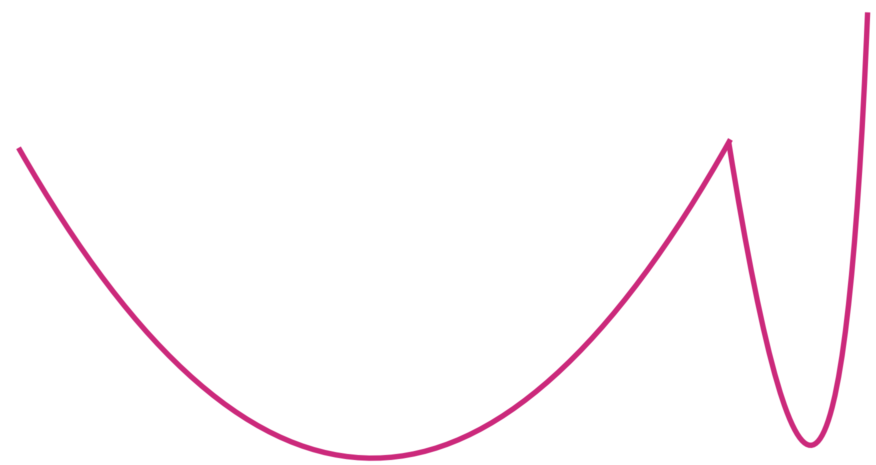
(f) C_4

Sharpness Metric

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

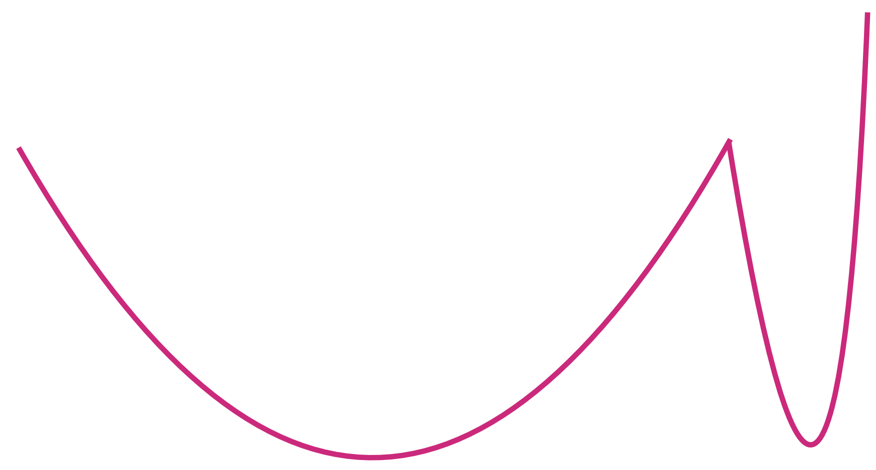
Sharpness Metric

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



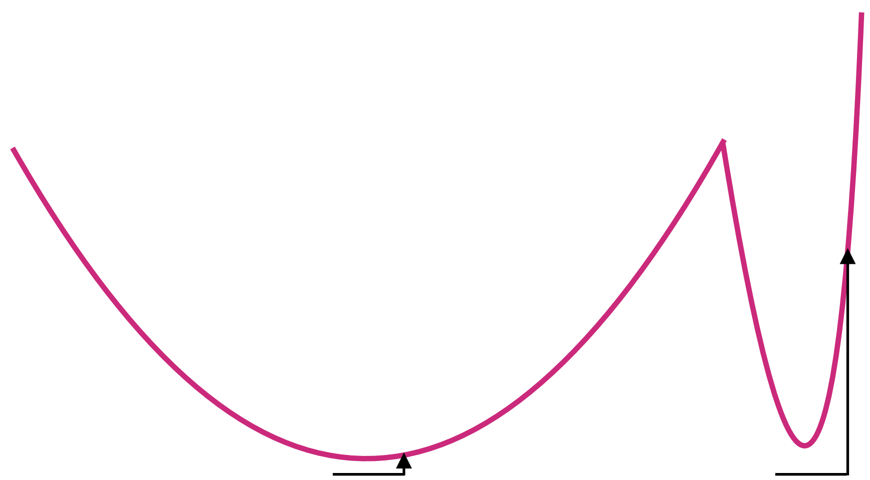
Sharpness Metric

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



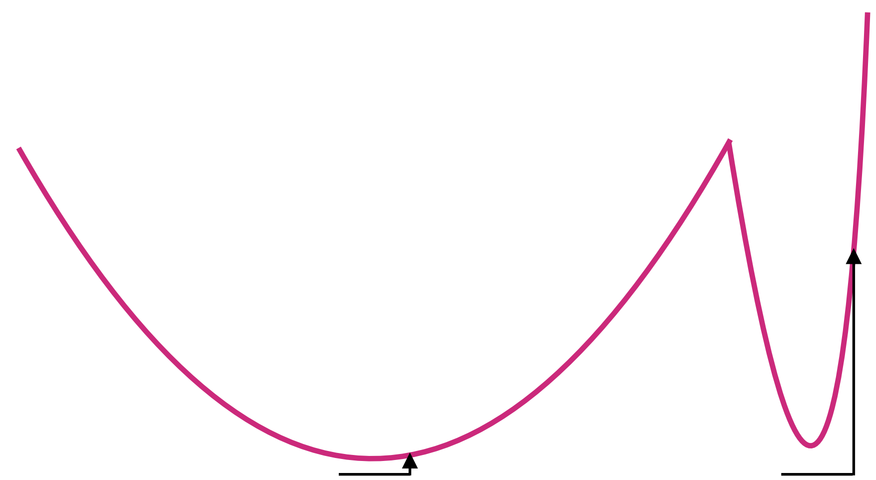
Sharpness Metric

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



Sharpness Metric

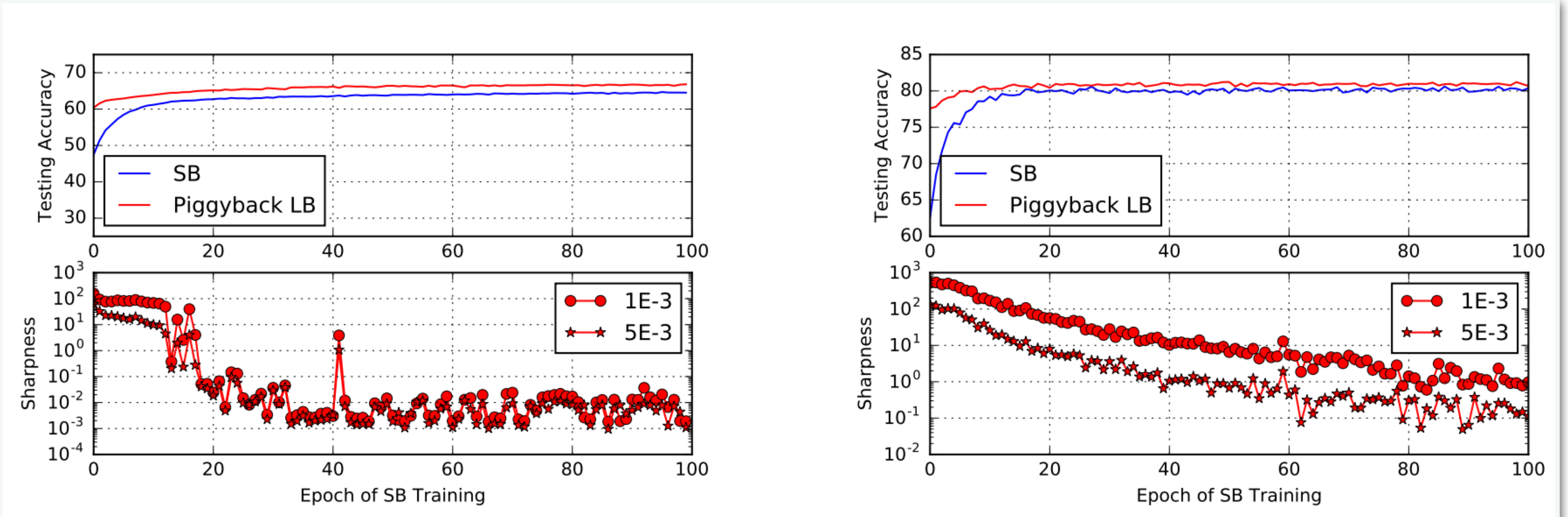
$$\phi_{x,f}(\epsilon, A) := \frac{(\max_{y \in \mathcal{C}_\epsilon} f(x + Ay)) - f(x)}{1 + f(x)} \times 100.$$



	$\epsilon = 10^{-3}$		$\epsilon = 5 \cdot 10^{-4}$	
	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

Other Observations

- Resilient to activation, BN, dropout, architecture etc.
- SB -> LB switch works, but needs to be timed *just right.*



(a) F_2 (b) C_1

- Several strategies (e.g., aggressive data augmentation and conservative training) helped close the generalization gap but not the sharpness gap.

Not an Optimization Issue

- Standard training (or “holy grail” L-BFGS/GD with line search) leads to good training, bad testing.
- Can’t be an optimization issue; we are *optimizing* the loss in every mathematical sense.

Not an Optimization Issue Not a Regularization Issue

- Standard training (or "holy grail" L-BFGS/GD with line search) leads to good training, bad testing.

- Can't be an optimization issue; we are *optimizing* the loss in every mathematical sense.

- Can't be a regularization issue; SB training trains/generalizes just fine for the same model.

- Need to explore the interplay between the model and the training dynamics; e.g., *"Train faster, generalize better: Stability of stochastic gradient descent"*.

Theoretical and Practical Developments

Theoretical and Practical Developments

1. “Sharp Minima Can Generalize For Deep Nets”

For ReLU networks, minima can be made arbitrarily sharp or flat. Insufficiency result.

2. “A Bayesian Perspective on Generalization and Stochastic Gradient Descent”

Bayesian evidence suggests poor generalizability of sharp minima and avoidance of SB SGD.

3. “Batch Size Matters: A Diffusion Approximation Framework on Nonconvex Stochastic Gradient Descent”

Looking at training from the lens of a multidimensional Ornstein-Uhlenbeck process, evidence that LB training ⇒ poor generalization.

4. “Three Factors Influencing Minima in SGD”, “Don't Decay the Learning Rate, Increase the Batch Size”

Connection between batch size, learning rate, size of the data set and noise scales.

Theoretical and Practical Developments

1. “Sharp Minima Can Generalize For Deep Nets”

   For ReLU networks, minima can be made arbitrarily sharp or flat. Insufficiency result.

2. “A Bayesian Perspective on Generalization and Stochastic Gradient Descent”

   Bayesian evidence suggests poor generalizability of sharp minima and avoidance of SB SGD.

3. “Batch Size Matters: A Diffusion Approximation Framework on Nonconvex Stochastic Gradient Descent”

   Looking at training from the lens of a multidimensional Ornstein-Uhlenbeck process, evidence that LB training $\Rightarrow$ poor generalization.

4. “Three Factors Influencing Minima in SGD”, “Don't Decay the Learning Rate, Increase the Batch Size”

   Connection between batch size, learning rate, size of the data set and noise scales.

- Through clever training regime changes (Hoffer et. al., Goyal et. al., You et. al., Akiba et. al.), possible to close down the generalization gap:
  - Warming up the learning rate (weakly equivalent to dynamic sampling).
  - Using distributed batch normalization.
  - Using corrections for momentum.
- Seems to change the basin that the optimizer moves towards.

Papers I Wish Were Written

- Why won’t standard or line search-based training methods work? Why do we need to warmup (LR/batch sizes)? What is warmup doing?

- Deep Learning is unlike other optimization problems. You get to select the f(x) that you wish to minimize. Choose it! Don’t hack the optimizer, hack the architecture.

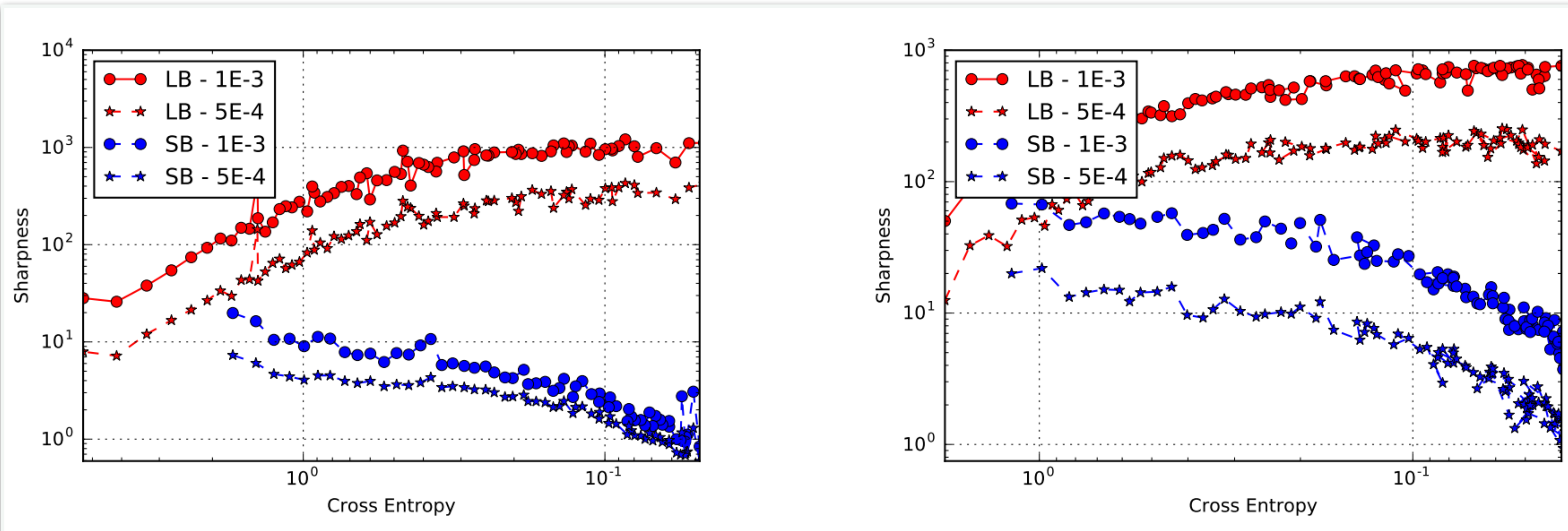
  - Possible self-selection; go back-to-basics for architecture design, regularization strategies and training regimes. RL Architecture search for LB?

- Stop training for 100s of epochs. A few should be all we need. Time is ripe for (quasi-) Newton methods. Also, variance reduced gradients.

- Take what we have learnt to Translation, Language Modeling, Seq2Seq.

Thank You!

Evolution of Sharpness



As we descend down the loss, SB (LB) sharpness keeps decreasing (increasing).