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

J. Nocedal

with

N. Keskar *Northwestern University*

D. Mudigere INTEL

P. Tang INTEL

M. Smelyanskiy INTEL

Initial Remarks

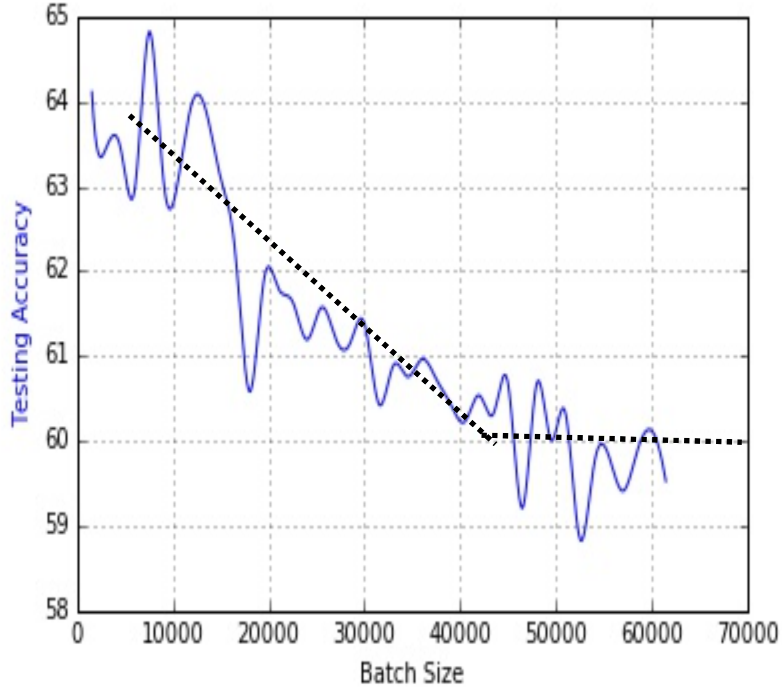
- SGD (and variants) is the method of choice
- Take another look at batch methods for training DNN
- Because they have the potential to parallelize
- Widely accepted that batch methods overfit
- Revisit this in the non-convex case of DNN with multiple minimizers

- Performed an exploration using ADAM where gradient sample increased from stochastic to batch regime
- Ran methods until no measurable progress is made in training
- Does the batch method converge to shallower minimizer?

- Testing Accuracy is lost with increase in batch size
- ADAM optimizer: 256 (small batch) v/s 10% (large batch)
- This behavior has been observed by others

Studied 6 network configurations

Testing Accuracy



Training and Testing Accuracy

SB: small batch LB: large batch

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

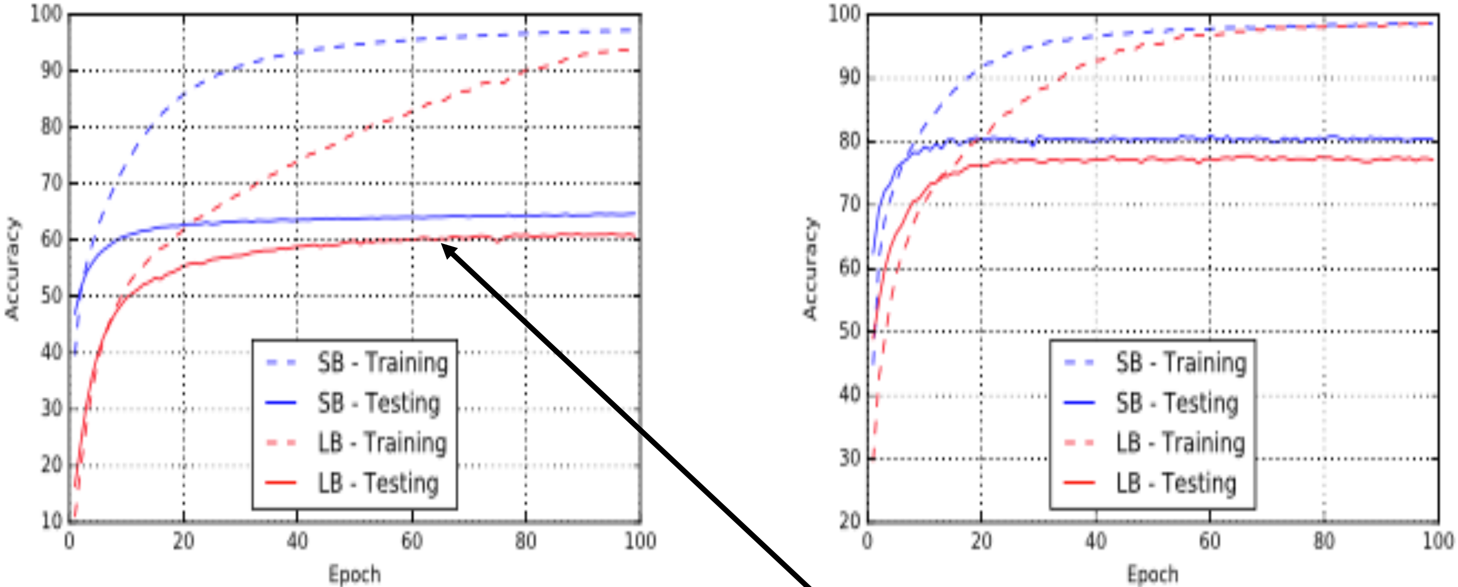
No Problems in Training!

Network configurations

Table 1: Network Configurations

Name	Network Type	Architecture	Data set
F_1	Fully Connected	Section B.1	MNIST (LeCun et al., 1998a)
F_2	Fully Connected	Section B.2	TIMIT (Garofolo et al., 1993)
C_1	(Shallow) Convolutional	Section B.3	CIFAR-10 (Krizhevsky & Hinton, 2009)
C_2	(Deep) Convolutional	Section B.4	CIFAR-10
C_3	(Shallow) Convolutional	Section B.3	CIFAR-100 (Krizhevsky & Hinton, 2009)
C_4	(Deep) Convolutional	Section B.4	CIFAR-100

Early stopping would not help large batch methods



(a) Network F_2 (b) Network C_1

Testing error for large batch method

Batch methods somehow do not employ information improperly

To be described mathematically **in this context!**

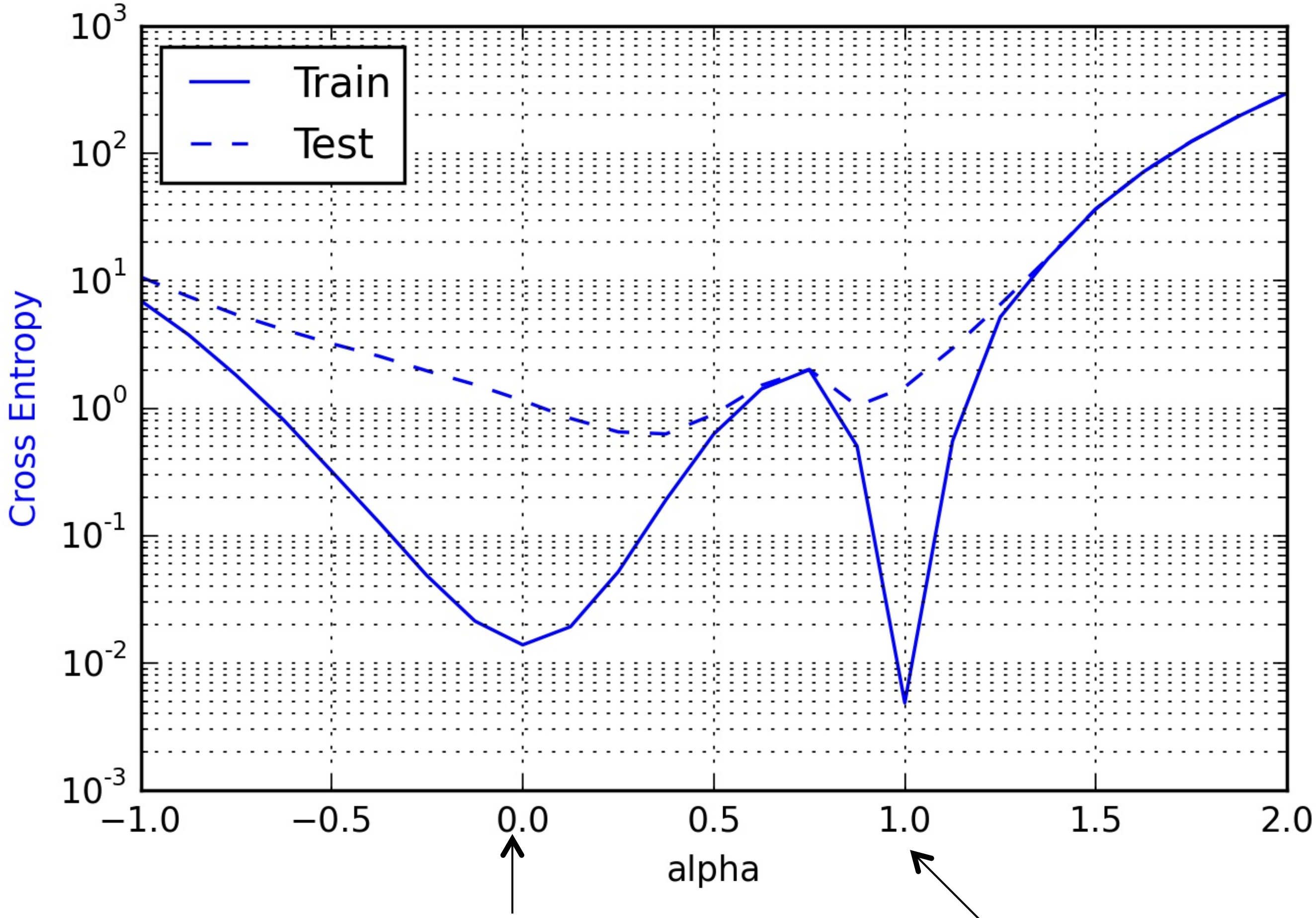
Methods converge to different types of minimizers

- Next: plot the geometry of the loss function along the line joining the small batch solution and large batch solution

- Plot the true loss and test functions

Goodfellow et al

W



Small batch solution

large batch solution

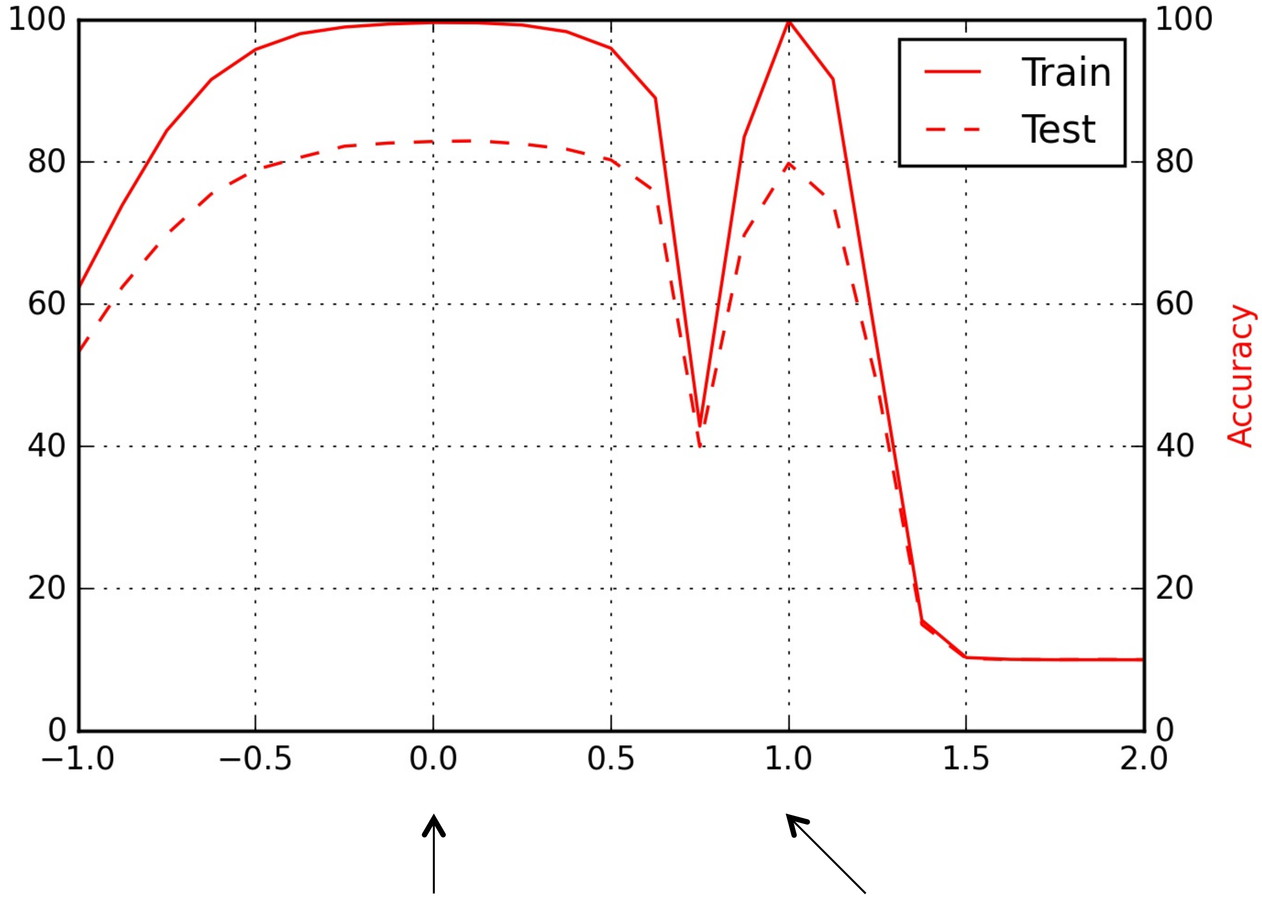
Convolutional neural net CIFAR-10

LeCun, private communication

SG: mini-batch of size 256

Batch: 10% of training set

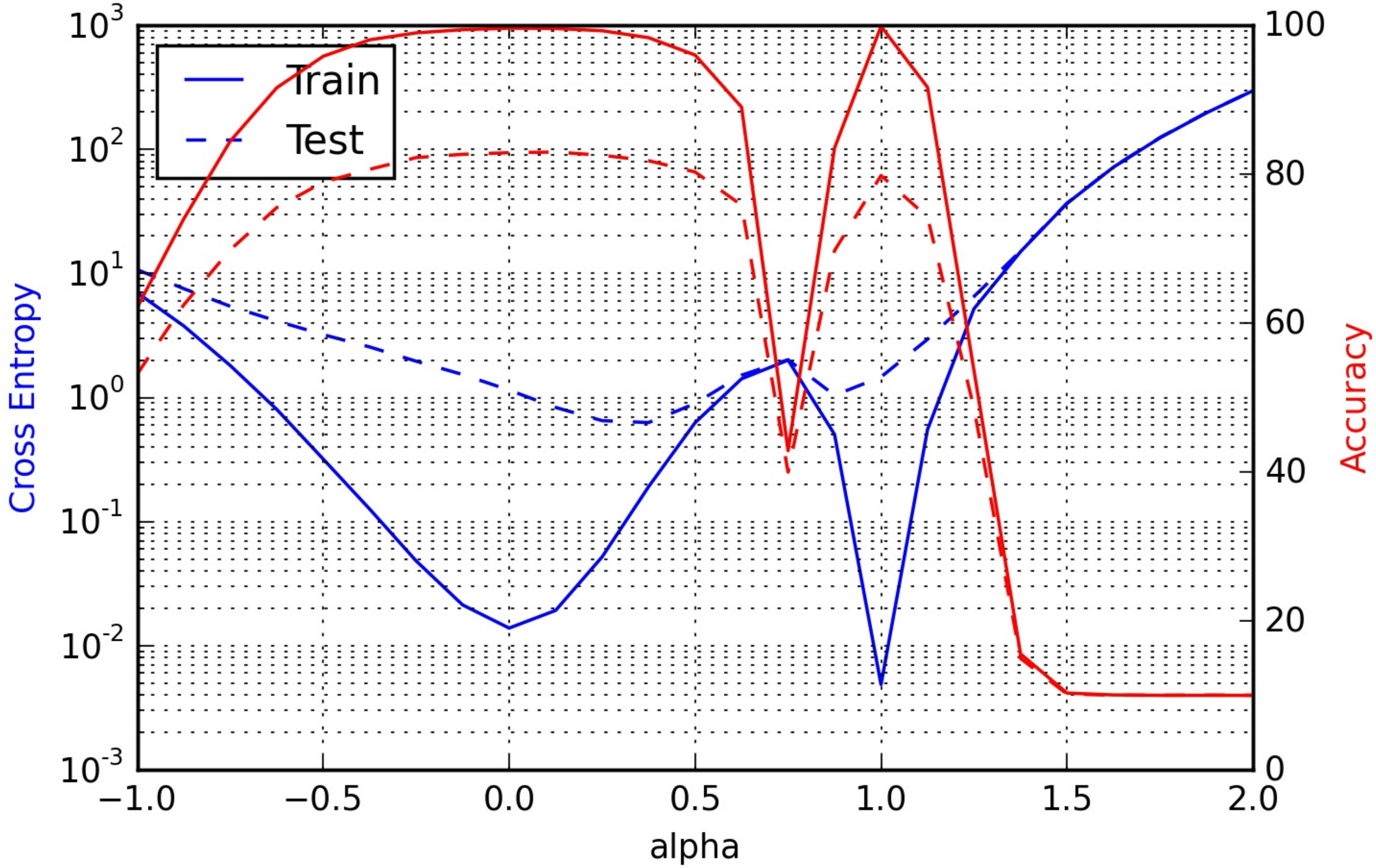
Accuracy: correct classification



SG solution

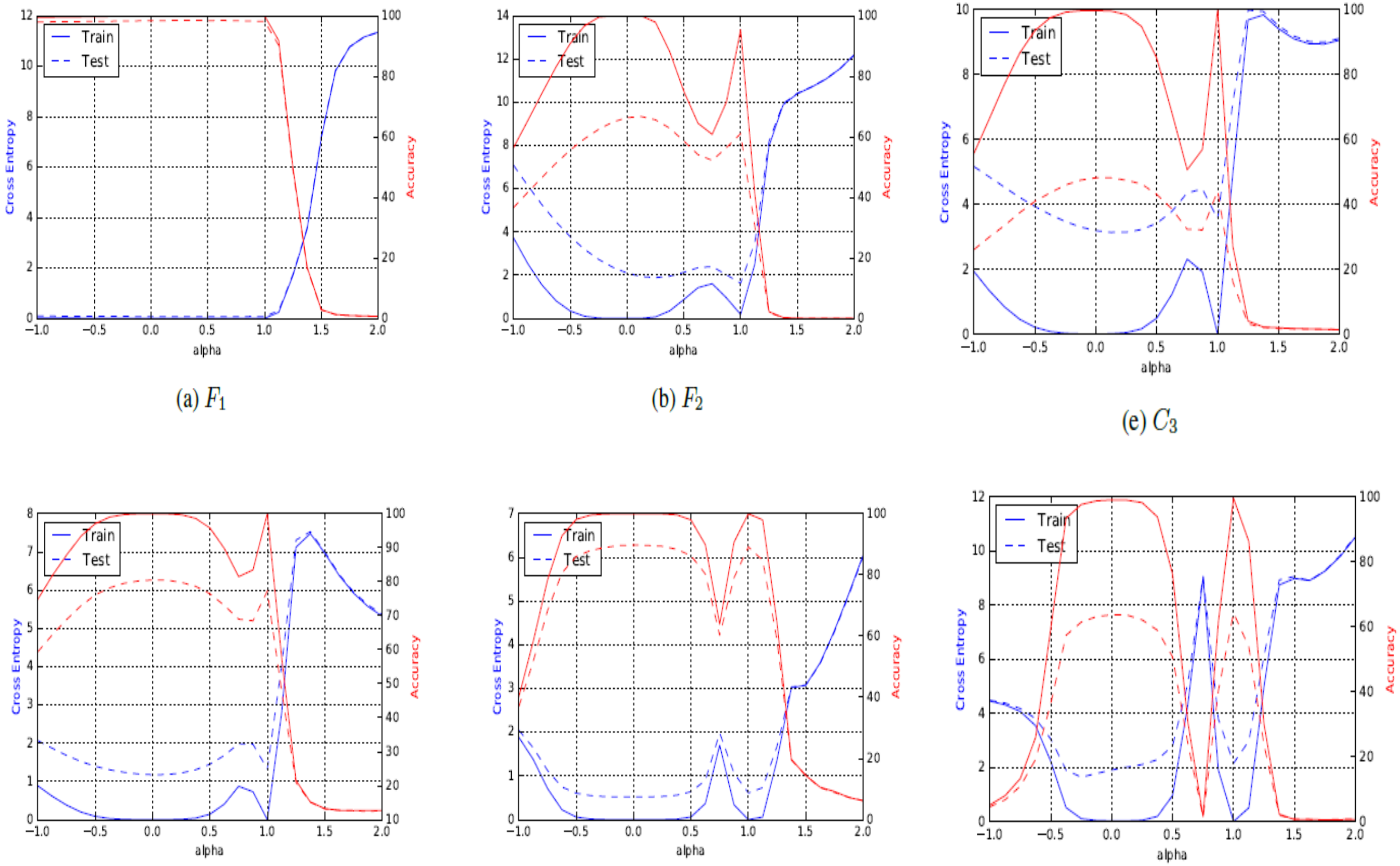
Batch solution

Combined



Train
Test
Cross Entropy
Accuracy
alpha
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
100
80
60
40
20
0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

We observe this over and over …



(a) F_1 (b) F_2 (e) C_3

(c) C_1 (d) C_2 (f) C_4

Has this been observed by others?

Hochreiter and Schmidhuber. Flat minima. 1997

What are Sharp and Wide Minima?

Volume. Free Energy. Robust Solution. Instead we use

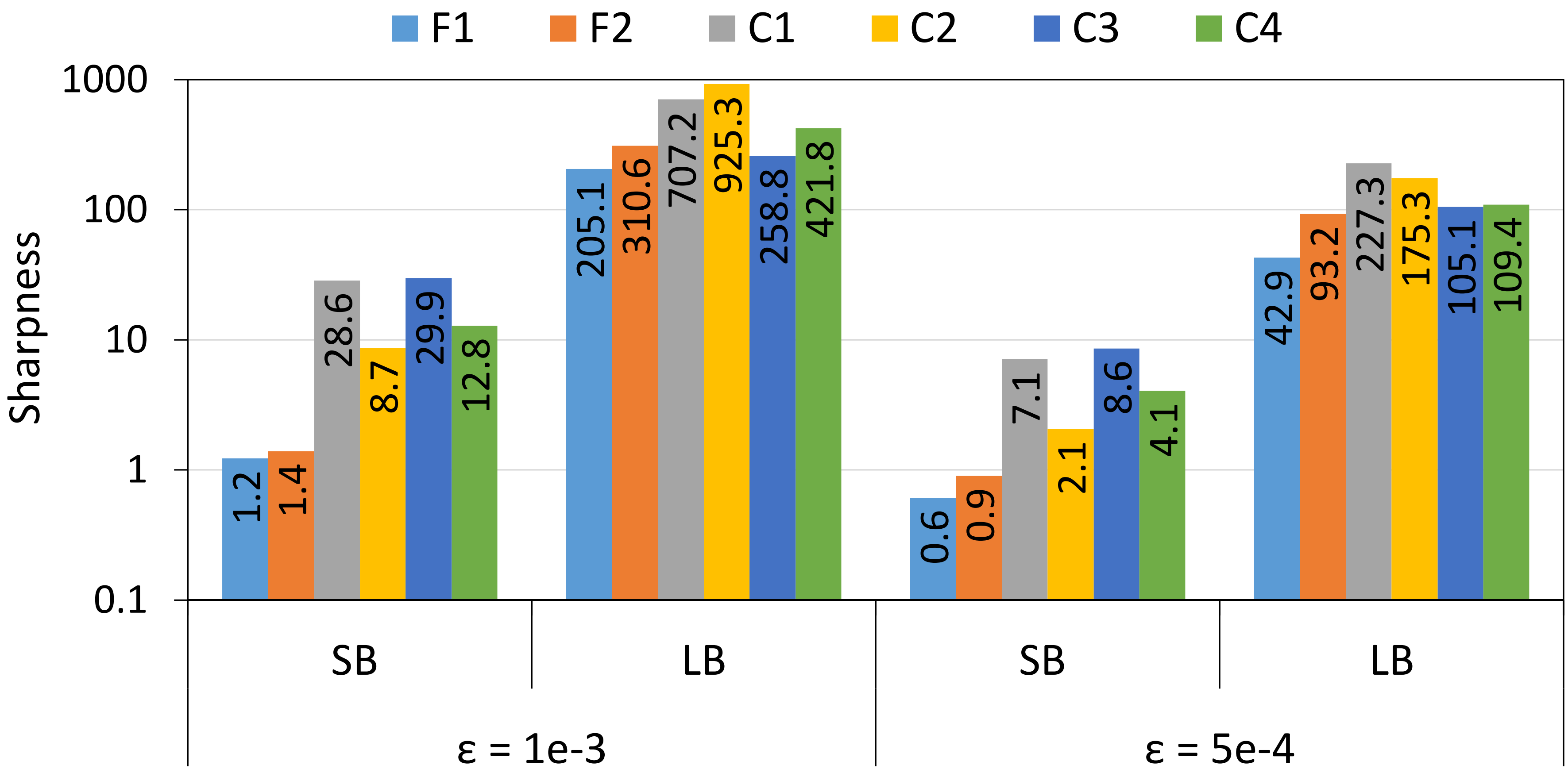
Given a parameter w^* and a box B of width ϵ centered at w^*, we define the sharpness of w^* as

$$max_{w \in B} \frac{f(w^* + w) - f(w^*)}{1 + f(w^*)}$$

1. Maximum sensitivity
2. Observed "sharp" solutions are "wide" in most of the space
3. Computed with an optimization solver (inexactly)
4. Verified through random sampling
5. Also minimized/samples in random subspaces

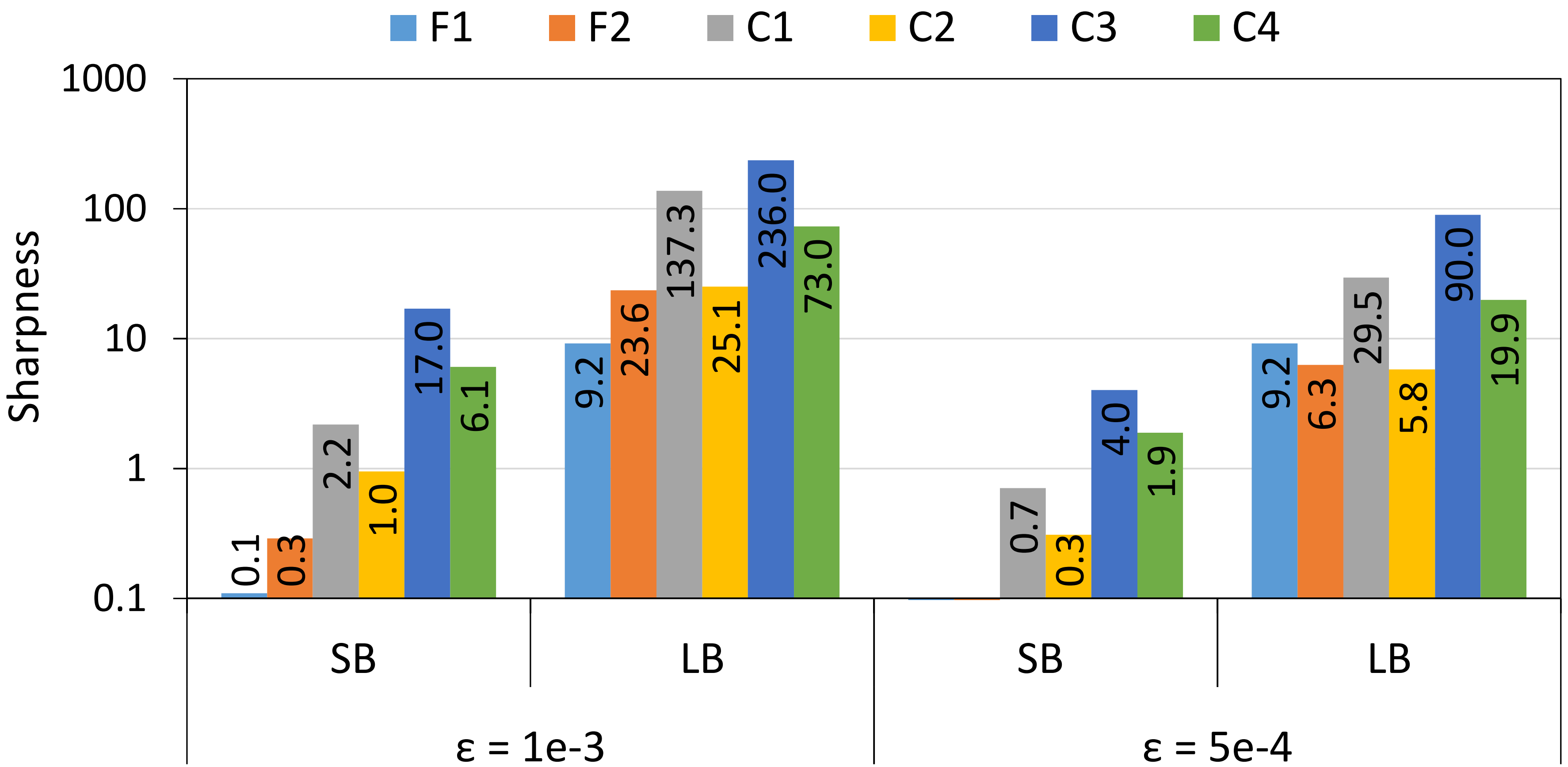
.

Sharpness: small batch solution SB large batch solution LB



Small batch solution large batch solution

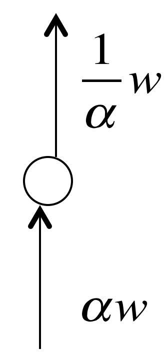
Sampling in a subsapce



Sharp and Wide Minima: an illusion?

1. It is tempting to conclude that convergence to sharp minima explains why batch methods do not generalize well
2. Perturbation analysis in parameter space refers to training problem
3. But geometry of loss function depends on the basis used in parameter space. One can alter it in various ways without changing prediction capability

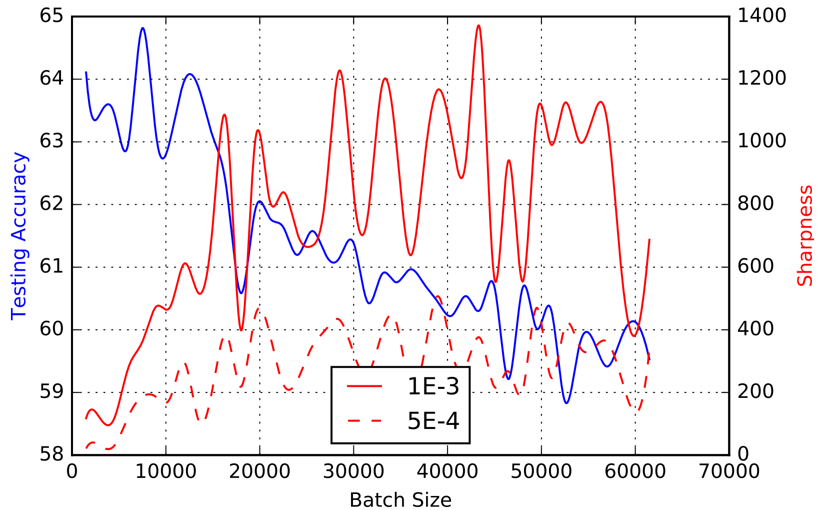
4. Dinh et al 2017 *Sharp Minima can generalize*:
5. construct two identical predictors; one
6. sharp minimum; the other not
7. Neyshabur et al: Path-SGD (2015)
8. Chaudhari et al. Entropy-sgd: Biasing gradient
9. descent into wide valleys 2016



Nevertheless our observations require an explanation

1. Sharpness grows as batch optimization iteration progresses
2. Controlled experiments: start with SGD and swtich to batch: can get trapped in sharp minima

.



Remarks

- Convergence to sharp/wide minima seems to be persistent
- Plausible: due to effect of noise in SGD and the fact that steplength is selected to give good testing error (noise adjustment)
- But it is not clear how to properly define sharp/wide minima so that they relate to generalization
- We need a mathematical explanation of the generalization properties of batch methods in the context of DNNs   (not convex case)
- And convergence of the optimization on training functions
- A batch method with good generalization properties could make use of parallel platforms