

# Contents

1 On large-batch training for deep learning: Generalization gap and sharp minima	1
1.1 Existing problems	1
1.2 Conclusions/Founding in this paper	1
1.3 Some background information	1
1.4 Experimental Study	2
1.4.1 How to measure sharpness of a minimizer	2
1.4.2 Comparison of the sharpness of the minimizers	2
1.4.3 minimizers and the starting point	4
1.5 References	4

## 1 On large-batch training for deep learning: Generalization gap and sharp minima

### 1.1 Existing problems

1. large batch methods over-fit the model.
2. large batch methods are attracted to saddle points.
3. large batch methods tend to zoom-in on the minimizer closest to the initial point.
4. small batch and large batch methods converge to qualitatively different minimizers with different properties.

### 1.2 Conclusions/Founding in this paper

- The generalization gap is not due to over-fitting or over-training.
- Due to the inherent noise in the gradient estimation:
  - large-batch methods tend to converge to sharp minimizers of the training and testing functions.
    - \* the sharp minimizers are characterized by a significant number of large positive eigenvalues in  $\nabla^2 f(x)$ .
    - \* sharp minimizers tend to generalize well.
  - small-batch methods consistently converge to flat minimizers.
    - \* flat minimizers characterized by having numerous small eigenvalues of  $\nabla^2 f(x)$ .
- The loss function landscape of deep neural networks is such that large-batch methods are attracted to region with sharp minimizers and are unable to escape basins of attraction of these minimizers.
- using a large-batch method that is warm-start with a small-batch method.

### 1.3 Some background information

- What is a flat minimizer and a sharp minimizer?
  - flat minimizer  $x^*$ : the function varies slowly in a relatively large neighborhood of  $x^*$ .
  - sharp minimizer  $x^*$ : the function increases rapidly in a small neighborhood of  $x^*$ .
- Many theoretical properties of SGD in small batch regime are known:
  1. converge to minimizers of strongly-convex functions and to stationary points for non-convex functions[1].

- 2. saddle-point avoidance[2].
- 3. robustness to input data[3].
- The loss function of deep learning models is fraught with many local minimizers, and many of these minimizers correspond to a similar loss function value.
  - both flat and sharp minimizer have very similar loss function values.

## 1.4 Experimental Study

### 1.4.1 How to measure sharpness of a minimizer

- sharpness of a minimizer can be characterized by the magnitude of the eigenvalues of  $\nabla^2 f(x)$ 
  - however, the computation cost is prohibitive.
- a computationally feasible metric: exploring a small neighborhood of a solution and computing the largest value that the function  $f$  can attain in that neighborhood.

space  $\mathbb{R}^n$  as well as in random manifolds. For that purpose, we introduce an  $n \times p$  matrix  $A$ , whose columns are randomly generated. Here  $p$  determines the dimension of the manifold, which in our experiments is chosen as  $p = 100$ .

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:

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

where  $A^+$  denotes the pseudo-inverse of  $A$ . Thus  $\epsilon$  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:

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

- pseudo-inverse

### 1.4.2 Comparison of the sharpness of the minimizers

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

- experiment 1:

Table 3: Sharpness of Minima in Full Space;  $\epsilon$  is defined in (3).

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

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

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

- experiment 2:

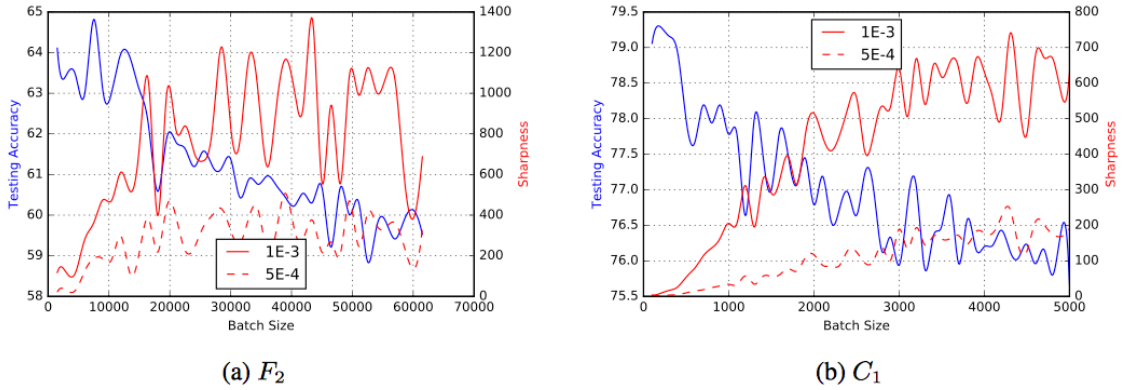


Figure 4: Testing Accuracy and Sharpness v/s Batch Size. The X-axis corresponds to the batch size used for training the network for 100 epochs, left Y-axis corresponds to the testing accuracy at the final iterate and right Y-axis corresponds to the sharpness of that iterate. We report sharpness for two values of  $\epsilon$ :  $10^{-3}$  and  $5 \cdot 10^{-4}$ .

- sharp minimizers identified in the experiments do not resemble a cone
  - sampling the loss function in neighborhood
  - the loss function rises steeply only along a small dimensional subspace.

### 1.4.3 minimizers and the starting point

- experimental method:
  1. train the network using Adam with batch size of 256 (small batch size), then we get 100 solutions.
  2. use these 100 solutions as start points and train the network use a target batch size.
- findings
  1. when warm-started with only a few initial epochs, the target batch method dose not yield a generalization improvment.
  2. after certain number of epochs of warm-starting, the accuracy improves and sharpness of the large-batch iterations drop.

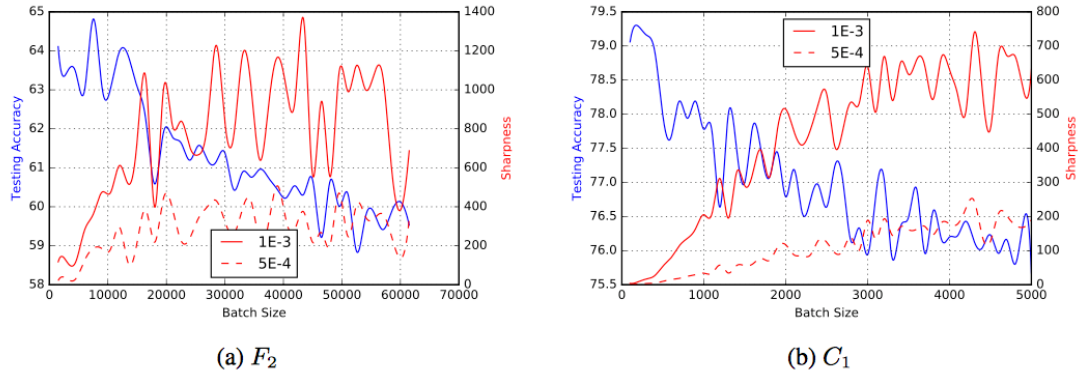


Figure 4: Testing Accuracy and Sharpness v/s Batch Size. The X-axis corresponds to the batch size used for training the network for 100 epochs, left Y-axis corresponds to the testing accuracy at the final iterate and right Y-axis corresponds to the sharpness of that iterate. We report sharpness for two values of  $\epsilon$ :  $10^{-3}$  and  $5 \cdot 10^{-4}$ .

the small batch method has ended its exploration phase and discovered a flat minimizer, the large batch method is then able to converge toward it, leading to a good testing accuracy.

### 1.5 References

1. Bottou L, Curtis F E, Nocedal J. Optimization methods for large-scale machine learning[J]. arXiv preprint arXiv:1606.04838, 2016.
2. Lee J D, Simchowitz M, Jordan M I, et al. Gradient descent converges to minimizers[J]. arXiv preprint arXiv:1602.04915, 2016.
3. Hardt M, Recht B, Singer Y. Train faster, generalize better: Stability of stochastic gradient descent[J]. arXiv preprint arXiv:1509.01240, 2015.