Large-Batch Training and Generalization Gap

Nitish Shirish Keskar
Salesforce Research
Slides: keskarnnitish.github.io
Deep Learning & SGD

\[
\min_{x \in \mathbb{R}^n} \ f(x) := \frac{1}{M} \sum_{i=1}^{M} f_i(x)
\]
Deep Learning & SGD

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

\[
x_{k+1} = x_k - \alpha_k \left( \frac{1}{|B_k|} \sum_{i \in B_k} \nabla f_i(x_k) \right)
\]
Deep Learning & SGD

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

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

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

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

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

- Large batch sizes \(\Rightarrow\) more concurrency \(\Rightarrow\) easier (data) parallelization \(\Rightarrow\) 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.
<table>
<thead>
<tr>
<th>Name</th>
<th>Training Accuracy</th>
<th>Testing Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SB</td>
<td>LB</td>
</tr>
<tr>
<td>$F_1$</td>
<td>99.66% ± 0.05%</td>
<td>99.92% ± 0.01%</td>
</tr>
<tr>
<td>$F_2$</td>
<td>99.99% ± 0.03%</td>
<td>98.95% ± 2.08%</td>
</tr>
<tr>
<td>$C_1$</td>
<td>99.89% ± 0.02%</td>
<td>99.66% ± 0.2%</td>
</tr>
<tr>
<td>$C_2$</td>
<td>99.99% ± 0.04%</td>
<td>99.99% ± 0.01%</td>
</tr>
<tr>
<td>$C_3$</td>
<td>99.56% ± 0.44%</td>
<td>99.88% ± 0.30%</td>
</tr>
<tr>
<td>$C_4$</td>
<td>99.10% ± 1.23%</td>
<td>99.57% ± 1.84%</td>
</tr>
<tr>
<td>Name</td>
<td>Training Accuracy</td>
<td>Testing Accuracy</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td>SB</td>
<td>LB</td>
</tr>
<tr>
<td>(F_1)</td>
<td>99.66% ± 0.05%</td>
<td>99.92% ± 0.01%</td>
</tr>
<tr>
<td>(F_2)</td>
<td>99.99% ± 0.03%</td>
<td>98.35% ± 2.08%</td>
</tr>
<tr>
<td>(C_1)</td>
<td>99.89% ± 0.02%</td>
<td>99.66% ± 0.2%</td>
</tr>
<tr>
<td>(C_2)</td>
<td>99.99% ± 0.04%</td>
<td>99.99% ± 0.01%</td>
</tr>
<tr>
<td>(C_3)</td>
<td>99.56% ± 0.44%</td>
<td>99.88% ± 0.30%</td>
</tr>
<tr>
<td>(C_4)</td>
<td>99.10% ± 1.23%</td>
<td>99.57% ± 1.84%</td>
</tr>
</tbody>
</table>

Network | LB size | Dataset  | SB  | LB | +LR |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alexnet</td>
<td>4096</td>
<td>ImageNet</td>
<td>57.10%</td>
<td>41.23%</td>
<td>53.25%</td>
</tr>
<tr>
<td>Alexnet</td>
<td>8192</td>
<td>ImageNet</td>
<td>57.10%</td>
<td>41.23%</td>
<td>53.25%</td>
</tr>
</tbody>
</table>

Hoffer et. al. (2017)
### Goyal et. al. (2017)

**Table: Training and Testing Accuracy**

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

### Hoffer et. al. (2017)

**Table: Network, Dataset, and Accuracy**

<table>
<thead>
<tr>
<th>Network</th>
<th>LB size</th>
<th>Dataset</th>
<th>SB</th>
<th>LB$^6$</th>
<th>+LR$^6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alexnet</td>
<td>4096</td>
<td>ImageNet</td>
<td>57.10%</td>
<td>41.23%</td>
<td>53.25%</td>
</tr>
<tr>
<td>Alexnet</td>
<td>8192</td>
<td>ImageNet</td>
<td>57.10%</td>
<td>41.23%</td>
<td>53.25%</td>
</tr>
<tr>
<td>F1 (Keskar et al., 2017)</td>
<td>MNIST</td>
<td>98.27%</td>
<td>97.05%</td>
<td>97.55%</td>
<td></td>
</tr>
<tr>
<td>C1 (Keskar et al., 2017)</td>
<td>Cifar10</td>
<td>87.80%</td>
<td>83.95%</td>
<td>86.15%</td>
<td></td>
</tr>
<tr>
<td>Resnet44 (He et al., 2016)</td>
<td>Cifar10</td>
<td>92.83%</td>
<td>86.10%</td>
<td>89.30%</td>
<td></td>
</tr>
<tr>
<td>VGG (Simonyan, 2014)</td>
<td>Cifar10</td>
<td>92.30%</td>
<td>84.1%</td>
<td>88.6%</td>
<td></td>
</tr>
<tr>
<td>C3 (Keskar et al., 2017)</td>
<td>Cifar100</td>
<td>61.25%</td>
<td>51.50%</td>
<td>57.38%</td>
<td></td>
</tr>
<tr>
<td>WResnet16-4 (Zagoruyko, 2016)</td>
<td>Cifar100</td>
<td>73.70%</td>
<td>68.15%</td>
<td>69.05%</td>
<td></td>
</tr>
</tbody>
</table>

### Goyal et. al. (2017)

**Table: Top-1 Error (%)**

<table>
<thead>
<tr>
<th></th>
<th>$k$</th>
<th>$n$</th>
<th>$kn$</th>
<th>$\eta$</th>
<th>top-1 error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline (single server)</td>
<td>8</td>
<td>32</td>
<td>256</td>
<td>0.1</td>
<td>23.60 ±0.12</td>
</tr>
<tr>
<td>no warmup, Figure 2a</td>
<td>256</td>
<td>32</td>
<td>8k</td>
<td>3.2</td>
<td>24.84 ±0.37</td>
</tr>
</tbody>
</table>
The “Why” Hypothesis
Flavors of Minima
The “Why” Hypothesis
Flavors of Minima

• LB training $\Rightarrow$ convergence to *sharp minima* of the loss.
The “Why” Hypothesis
Flavors of Minima

• LB training $\Rightarrow$ 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

\[ x_{sb} + (1-\alpha)x_{sb} \]

\[ \alpha = 0 \rightarrow x_{lb} \]

\[ \alpha = 1 \rightarrow x_{lb} \]
Parametric Plots

\[ x_{lb} = \alpha x_{lb} + (1-\alpha)x_{sb} \]

- \( \alpha = 0 \)
- \( \alpha = 1 \)
- \( \alpha = 2 \)
Deep ConvNet on CIFAR10
Deep ConvNet on CIFAR10

![Graph showing Cross Entropy vs. alpha for SB and LB Solutions]
Deep ConvNet on CIFAR10

![Graph 1](graph1.png)

- SB Solution
- LB Solution

![Graph 2](graph2.png)

- SB Solution
- LB Solution
Sharpness Metric

\[ \phi_{x,f}(\epsilon, A) := \frac{\max_{y \in C_\epsilon} f(x + Ay) - f(x)}{1 + f(x)} \times 100. \]
Sharpness Metric

\[ \phi_{x,f}(\varepsilon, A) := \frac{\max_{y \in \mathcal{C}_\varepsilon} f(x + Ay) - f(x)}{1 + f(x)} \times 100. \]
Sharpness Metric

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

\[ \phi_{x,f}(\epsilon, A) := \frac{\left( \max_{y \in C_{x}} f(x + Ay) \right) - f(x)}{1 + f(x)} \times 100. \]
Sharpness Metric

\[
\phi_{x,f}(\epsilon, A) := \frac{\max_{y \in C_\epsilon} f(x + Ay)) - f(x)}{1 + f(x)} \times 100.
\]

<table>
<thead>
<tr>
<th></th>
<th>(\epsilon = 10^{-3})</th>
<th></th>
<th>(\epsilon = 5 \cdot 10^{-4})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SB</td>
<td>LB</td>
<td>SB</td>
</tr>
<tr>
<td>(F_1)</td>
<td>1.23 ± 0.83</td>
<td>205.14 ± 69.52</td>
<td>0.61 ± 0.27</td>
</tr>
<tr>
<td>(F_2)</td>
<td>1.39 ± 0.02</td>
<td>310.64 ± 38.46</td>
<td>0.90 ± 0.05</td>
</tr>
<tr>
<td>(C_1)</td>
<td>28.58 ± 3.13</td>
<td>707.23 ± 43.04</td>
<td>7.08 ± 0.88</td>
</tr>
<tr>
<td>(C_2)</td>
<td>8.68 ± 1.32</td>
<td>925.32 ± 38.29</td>
<td>2.07 ± 0.86</td>
</tr>
<tr>
<td>(C_3)</td>
<td>29.85 ± 5.98</td>
<td>258.75 ± 8.96</td>
<td>8.56 ± 0.99</td>
</tr>
<tr>
<td>(C_4)</td>
<td>12.83 ± 3.84</td>
<td>421.84 ± 36.97</td>
<td>4.07 ± 0.87</td>
</tr>
</tbody>
</table>
Other Observations

• Resilient to activation, BN, dropout, architecture etc.

• SB -> LB switch works, but needs to be timed *just right.*

• 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.

   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.

   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).