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

**Testing Accuracy**

Studied 6 network configurations
## Training and Testing Accuracy

SB: small batch    LB: large batch

<table>
<thead>
<tr>
<th>Network 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>$99.66% \pm 0.05%$</td>
<td>$99.92% \pm 0.01%$</td>
</tr>
<tr>
<td>$F_2$</td>
<td>$99.99% \pm 0.03%$</td>
<td>$98.35% \pm 2.08%$</td>
</tr>
<tr>
<td>$C_1$</td>
<td>$99.89% \pm 0.02%$</td>
<td>$99.66% \pm 0.2%$</td>
</tr>
<tr>
<td>$C_2$</td>
<td>$99.99% \pm 0.04%$</td>
<td>$99.99 \pm 0.01%$</td>
</tr>
<tr>
<td>$C_3$</td>
<td>$99.56% \pm 0.44%$</td>
<td>$99.88% \pm 0.30%$</td>
</tr>
<tr>
<td>$C_4$</td>
<td>$99.10% \pm 1.23%$</td>
<td>$99.57% \pm 1.84%$</td>
</tr>
</tbody>
</table>

No Problems in Training!
## Network Configurations

<table>
<thead>
<tr>
<th>Name</th>
<th>Network Type</th>
<th>Architecture</th>
<th>Data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$</td>
<td>Fully Connected</td>
<td>Section B.1</td>
<td>MNIST (LeCun et al., 1998a)</td>
</tr>
<tr>
<td>$F_2$</td>
<td>Fully Connected</td>
<td>Section B.2</td>
<td>TIMIT (Garofolo et al., 1993)</td>
</tr>
<tr>
<td>$C_1$</td>
<td>(Shallow) Convolutional</td>
<td>Section B.3</td>
<td>CIFAR-10 (Krizhevsky &amp; Hinton, 2009)</td>
</tr>
<tr>
<td>$C_2$</td>
<td>(Deep) Convolutional</td>
<td>Section B.4</td>
<td>CIFAR-10</td>
</tr>
<tr>
<td>$C_3$</td>
<td>(Shallow) Convolutional</td>
<td>Section B.3</td>
<td>CIFAR-100 (Krizhevsky &amp; Hinton, 2009)</td>
</tr>
<tr>
<td>$C_4$</td>
<td>(Deep) Convolutional</td>
<td>Section B.4</td>
<td>CIFAR-100</td>
</tr>
</tbody>
</table>
Early stopping would not help large batch methods

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
What is going on?

Gradient method "over-fits"

We need to back-up:

Define setting of supervised training

Small batch solution                           large batch solution

SG: mini-batch of size 256        Batch: 10% of training set

Convolutional neural net
CIFAR-10

LeCun, private communication
Accuracy: correct classification

![Graph showing accuracy for SG solution and Batch solution](image-url)
Combined

![Graph showing the relationship between cross-entropy and accuracy with varying alpha values. The graph plots cross-entropy on the y-axis and accuracy on the right y-axis, with alpha on the x-axis. Two curves are shown: one for training (solid blue line) and one for testing (dashed red line). The graph demonstrates the trade-off between cross-entropy and accuracy as alpha changes.]
We observe this over and over …

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 $\epsilon$ 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

![Bar chart showing sharpness values for SB and LB in two different ε values: 1e-3 and 5e-4. The chart compares different values for F1, F2, C1, C2, C3, and C4.](chart-image-url)
Sampling in a subspace

![Bar chart showing sharpness values for different subspaces and error tolerances.]

- **SB** and **LB** represent different subspaces.
- **F1**, **F2**, **C1**, **C2**, **C3**, and **C4** represent different conditions or parameters.
- The y-axis represents sharpness values.
- The chart includes two error tolerances: \(\varepsilon = 1e^{-3}\) and \(\varepsilon = 5e^{-4}\).

**Values:**
- For **SB** and \(\varepsilon = 1e^{-3}\): 0.1, 0.3, 2.2, 1.0, 17.0, 6.9.
- For **LB** and \(\varepsilon = 1e^{-3}\): 9.2, 23.6, 137.3, 25.1, 236.0, 73.0.
- For **SB** and \(\varepsilon = 5e^{-4}\): 0.7, 0.3, 4.0, 1.9, 6.3, 29.5.
- For **LB** and \(\varepsilon = 5e^{-4}\): 9.2, 6.3, 5.8, 90.0, 19.9.
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


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 switch 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.