Momentum in Stochastic Gradient Descent and Neural Architecture Design

Bao Wang
Department of Mathematics, UCLA

Joint work with Tan Nguyen, Rich Baraniuk, Andrea Bertozzi, and Stan Osher
Deep Learning (DL)

\[ \text{DL} = \text{Big Data} + \text{Deep Nets} + \text{SGD} + \text{HPC} \]
Deep Learning: Revolution in Technology

**Face ID**

**Autonomous Cars**

**Alpha Go**

**Machine Translation**

How are you?

Τι κάνετε;
Deep Learning: Revolution in Science

Protein Structure Prediction

Drug Discovery

Molecular Generation

Material Design
Momentum in SGD

Accelerate convergence
Better generalizable models

Momentum in Deep Nets Design

Principled approach
Easier to train & Better generalizability
Momentum in Optimization
Machine Learning – Empirical Risk Minimization (ERM)

Consider training a machine learning model:

$$y = g(x, w), w \in \mathbb{R}^d,$$

which is parameterized by $w$, $(x, y)$ is an input data-predicted label pair.

$w$ is usually learned by solving the following ERM problem:

$$\min_w f(w) := \frac{1}{N} \sum_{i=1}^{N} f_i(w) := \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(g(x_i, w), y_i),$$

where $\mathcal{L}$ is the loss between the predicted label $\hat{y}_i$ and the ground-truth label $y_i$.

For classification, typically we have:

$$\mathcal{L}(\hat{y}_i, y_i) = - \sum_{j=1}^{M} y_i^j \log(p_i^j),$$

where $p_i^j$ is the predicted probability that $y_i$ is belong to $y_i^j$’s class.

$f(w)$ can be highly nonconvex, and $d \sim 10^{10}$, $N \sim 10^9$. 
Suppose $f(w)$ is convex and $L$-smooth. Start from $w_0$, and perform:

$$w_k = w_{k-1} - s \nabla f(w_{k-1}), \text{ and } s = 1/L.$$ 

Gradient descent has a convergence rate $O(1/k)$ for convex smooth functions.

Consider

$$\min_w f(w) = \frac{1}{2} w^T L w - w^T b,$$

where

$$L = \begin{pmatrix}
2 & -1 & 0 & \cdots & 0 & 0 \\
-1 & 2 & -1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & 0 & -1 & 2 & -1 \\
0 & 0 & \cdots & 0 & -1 & 2
\end{pmatrix}_{1000 \times 1000},$$

and $b$ is a 1000-dim vector whose first entry is 1 and all the other entries are 0.

A. Cauchy, 1847
Convex Optimization – Gradient Descent
Convex Optimization – Gradient Descent + Momentum

\[ \mathbf{v}_k = \mathbf{w}_{k-1} - s \nabla f(\mathbf{w}_{k-1}), \]

\[ \mathbf{w}_k = \mathbf{v}_k + \mu (\mathbf{v}_k - \mathbf{v}_{k-1}). \]

\( O(1/k) \) convergence rate!
Heavy Ball

\[ w_k = w_{k-1} - s \nabla f(w_{k-1}) + \mu (w_{k-1} - w_{k-2}), \]

\( O(1/k) \) convergence rate!

B. Polyak, 1964
Why momentum works

Figure: Top: no momentum; Bottom: with momentum.

G. Goh, Why momentum really works. Distill, 2017
Other Variants (Look-ahead)

Let \( u_k := v_k - v_{k-1} \) and \( g_k := \nabla f(v_{k-1} + su_{k-1}) \).

Sutskever et al. 2013
\[
\begin{align*}
v_k &= \mu v_{k-1} + sg_k, \\
w_k &= w_{k-1} - v_k.
\end{align*}
\]

PyTorch
\[
\begin{align*}
v_k &= \mu v_{k-1} + g_k, \\
w_k &= w_{k-1} - s v_k.
\end{align*}
\]
Convex Optimization – Nesterov Accelerated Gradient (NAG)

\[ \mathbf{v}_k = \mathbf{w}_{k-1} - s \nabla f(\mathbf{w}_{k-1}), \]
\[ \mathbf{w}_k = \mathbf{v}_k + \frac{k-1}{k+2} (\mathbf{v}_k - \mathbf{v}_{k-1}). \]

\( O(1/k^2) \) convergence rate!
Convex Optimization – Adaptive Restart NAG (ARNAG)

\[ \mathbf{v}_k = \mathbf{w}_{k-1} - s\nabla f(\mathbf{w}_{k-1}), \]

\[ \mathbf{w}_k = \mathbf{v}_k + \frac{t - 1}{t + 2}(\mathbf{v}_k - \mathbf{v}_{k-1}), \quad t \text{ increase by 1 if objective decay, else } t = 1. \]

\[ O(e^{-\alpha k}) \] convergence rate with an extra sharpness assumption!

Sharpness: \( \frac{\mu}{r} d(x, X^*)^r < f(x) - f^*, \quad \mu > 0, \ r > 1. \)

![Graph showing the convergence of different optimization methods](image)
Convex Optimization – Scheduled Restart NAG (SRNAG)

\[ \mathbf{v}_k = \mathbf{w}_{k-1} - s \nabla f(\mathbf{w}_{k-1}), \]

\[ \mathbf{w}_k = \mathbf{v}_k + \frac{t - 1}{t + 2} (\mathbf{v}_k - \mathbf{v}_{k-1}), \quad t \text{ increase by 1 within some iterations, else } t = 1. \]

\( O(e^{-\beta k}) \) convergence rate with an extra sharpness assumption!
What If We Do Not Have Exact Gradient?

In ERM,

$$\min_w f(w) := \frac{1}{N} \sum_{i=1}^{N} f_i(w) := \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(g(x_i, w), y_i),$$

when $N \gg 1$, compute $\nabla f(w)$ will be very expensive.

Stochastic Gradient:

$$\nabla f(w) \approx \frac{1}{n} \sum_{i=1}^{n} f_i(w), \text{ with } [n] \subset [N] \text{ and } n \ll N.$$

Can NAG still accelerate convergence with Stochastic Gradient?
Consider
\[
\min_w f(w) = \frac{1}{2} w^T L w - w^T b,
\]
where
\[
L = \begin{pmatrix}
2 & -1 & 0 & \cdots & 0 & 0 \\
-1 & 2 & -1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & 0 & -1 & 2 & -1 \\
0 & 0 & \cdots & 0 & -1 & 2
\end{pmatrix}_{1000 \times 1000},
\]
and \(b\) is a 1000-dim vector whose first entry is 1 and all the other entries are 0.

Gaussian Noise Corrupted Gradient:
\[
\nabla f(w) = Lw - b + n, \quad n \sim \mathcal{N}(0, (0.1^{\left\lfloor k/100 \right\rfloor + 1})^2).
\]
A Motivating Example – Gaussian Noise Corrupted Gradient – Case I

The graph shows the convergence of different optimization algorithms over iterations. The y-axis represents the function value difference $f(x^k) - f(x^*)$ on a logarithmic scale, while the x-axis represents the number of iterations (k). The graph compares the performance of GD, GD + Momentum, NAG, ARNAG, and SRNAG algorithms. The x-axis is marked with the number of iterations, with values ranging from $10^{-4}$ to $10^0$. The y-axis is marked with different logarithmic scales to show the relative performance of the algorithms.
A Motivating Example – Gaussian Noise Corrupted Gradient – Case II

Consider

$$\min_w f(w) = \frac{1}{2} w^T L w - w^T b,$$

where

$$L = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & 0 & \cdots & 0 & -1 & 2 \end{pmatrix}_{1000 \times 1000},$$

and $b$ is a 1000-dim vector whose first entry is 1 and all the other entries are 0.

Gaussian Noise Corrupted Gradient:

$$\nabla f(w) = Lw - b + n, \quad n \sim \mathcal{N}(0, 0.001^2).$$
A Motivating Example – Gaussian Noise Corrupted Gradient – Case II

- GD
- GD + Momentum
- NAG
- ARNAG
- SRNAG
A Motivating Example – Logistic Regression – Case III

Figure: Training loss of logistic regression for MNIST classification.
NAG:
Given a convex function \( f : \mathbb{R}^d \rightarrow \mathbb{R} \) with smoothness parameter \( L \). Given any point \( w \in \mathbb{R}^d \) an exact first order oracle returns a pair \((f_L(x), g_L(w)) \in \mathbb{R} \times \mathbb{R}^d\) so that for all \( u \in \mathbb{R}^d \), we have
\[
0 \leq f(u) - (f_L(w) + \langle g_L(w), u - w \rangle) \leq \frac{L}{2} \|u - w\|^2.
\]

The first inequality follows from convexity and the second from the smoothness condition.
An inexact oracle returns for any given point \( w \in \mathbb{R}^d \) a pair \((f_{\delta,L}(x), g_{\delta,L}(w)) \in \mathbb{R} \times \mathbb{R}^d\) so that for all \( u \in \mathbb{R}^d \) we have
\[
0 \leq f(u) - (f_{\delta,L}(w) + \langle g_{\delta,L}(w), u - w \rangle) \leq \frac{L}{2} \|u - w\|^2 + \delta.
\]

It has the same picture as before except now there is some \( \delta \) slack between the linear and parabola approximations.

Theorem
Given access to \( \delta \)-inexact first-order oracle GD spits out a point \( w^k \) after \( k \) steps so that
\[
f(w^k) - f(w^*) \leq O(L/k) + \delta.
\]

NAG on the other hand gives
\[
f(w^k) - f(w^*) \leq O(L/k^2) + O(k\delta).
\]

Adaptive Restart NAG: restart too often, degenerates to GD.

SRNAG for Deep Learning – ResNet for CIFAR10 and CIFAR100 Classification

[Images of airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck]

[Graph showing loss in log scale over epochs for SRSGD and SGD]
<table>
<thead>
<tr>
<th>Network</th>
<th># Params</th>
<th>SGD (baseline)</th>
<th>SRSGD (linear)</th>
<th>SRSGD (exponential)</th>
<th>Improvement(linear/exponential)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PreResNet-110</td>
<td>1.1M</td>
<td>94.75 ± 0.14 (93.63 (He et al., 2016))</td>
<td>95.07 ± 0.13</td>
<td>95.00 ± 0.47</td>
<td>0.32/0.25</td>
</tr>
<tr>
<td>PreResNet-290</td>
<td>3.0M</td>
<td>94.95 ± 0.23</td>
<td>95.63 ± 0.15</td>
<td>95.50 ± 0.18</td>
<td>0.68/0.55</td>
</tr>
<tr>
<td>PreResNet-470</td>
<td>4.9M</td>
<td>95.08 ± 0.10</td>
<td>95.82 ± 0.09</td>
<td>95.51 ± 0.19</td>
<td>0.74/0.43</td>
</tr>
<tr>
<td>PreResNet-650</td>
<td>6.7M</td>
<td>95.13 ± 0.14</td>
<td>96.00 ± 0.07</td>
<td>95.60 ± 0.13</td>
<td>0.87/0.47</td>
</tr>
<tr>
<td>PreResNet-830</td>
<td>8.6M</td>
<td>95.00 ± 0.23</td>
<td>95.91 ± 0.10</td>
<td>95.71 ± 0.13</td>
<td>0.91/0.71</td>
</tr>
<tr>
<td>PreResNet-1001</td>
<td>10.3M</td>
<td>95.16 ± 0.19 (95.08 (He et al., 2016))</td>
<td>96.13 ± 0.07</td>
<td>95.87 ± 0.10</td>
<td>0.97/0.71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network</th>
<th># Params</th>
<th>SGD (baseline)</th>
<th>SRSGD (linear)</th>
<th>SRSGD (exponential)</th>
<th>Improvement(linear/exponential)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PreResNet-110</td>
<td>1.2M</td>
<td>76.25 ± 0.20</td>
<td>76.51 ± 0.23</td>
<td>76.50 ± 0.39</td>
<td>0.26/0.25</td>
</tr>
<tr>
<td>PreResNet-290</td>
<td>3.0M</td>
<td>78.22 ± 0.21</td>
<td>78.51 ± 0.27</td>
<td>78.42 ± 0.20</td>
<td>0.29/0.20</td>
</tr>
<tr>
<td>PreResNet-470</td>
<td>4.9M</td>
<td>78.57 ± 0.30</td>
<td>79.29 ± 0.32</td>
<td>79.36 ± 0.18</td>
<td>0.72/0.79</td>
</tr>
<tr>
<td>PreResNet-650</td>
<td>6.7M</td>
<td>78.73 ± 0.14</td>
<td>79.64 ± 0.25</td>
<td>79.59 ± 0.21</td>
<td>0.91/0.86</td>
</tr>
<tr>
<td>PreResNet-830</td>
<td>8.6M</td>
<td>78.81 ± 0.22</td>
<td>79.78 ± 0.23</td>
<td>79.80 ± 0.25</td>
<td>0.97/0.99</td>
</tr>
<tr>
<td>PreResNet-1001</td>
<td>10.4M</td>
<td>79.13 ± 0.20 (77.29 (He et al., 2016))</td>
<td>80.25 ± 0.11</td>
<td>80.47 ± 0.19</td>
<td>1.12/1.34</td>
</tr>
</tbody>
</table>

**Figure:** Top: CIFAR10; Bottom: CIFAR100
SRNAG for Deep Learning – ResNet for ImageNet Classification
<table>
<thead>
<tr>
<th>Network</th>
<th># Params</th>
<th>SGD</th>
<th>SRSGD</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td>25.56M</td>
<td>75.83</td>
<td>76.24</td>
<td>0.41</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>44.55M</td>
<td>77.6</td>
<td>78.06</td>
<td>0.44</td>
</tr>
<tr>
<td>ResNet-152</td>
<td>60.19M</td>
<td>78.10</td>
<td>78.71</td>
<td>0.61</td>
</tr>
<tr>
<td>ResNet-200</td>
<td>64.67M</td>
<td>77.82</td>
<td>78.92</td>
<td>1.10</td>
</tr>
</tbody>
</table>
Image Classification

Figure: Error vs. depth of ResNet.
Momentum in Deep Neural Nets
Rethinking Deep Residual Nets (ResNets)

Residual mapping: \( x_{k+1} = x_k + f(x_k, w_k) \).

How about \( x_{k+1} = x_k - \mu f(x_k, w_k) \)?

What if

\[
y_{k+1} = x_k - \mu f(x_k, w_k),
\]

\[
x_{k+1} = y_{k+1} + \nu(y_{k+1} - y_k)?
\]
ResNet vs. Gradient Descent

Gradient descent

\[ x_{k+1} = x_k - \mu \nabla f(x_k), \]

and we have

\[ \lim_{k \to \infty} ||\nabla f(x_k)||_2 = 0. \]
Deep Momentum Nets

Residual block

\[ x_k \xrightarrow{f_{\phi}} x_{k+1} \]

Momentum block

\[ x_k \xrightarrow{f_{\phi}} 1 + \mu \xrightarrow{v_{k+1}} x_{k+1} \]

\[ -\mu = -\frac{k}{k+3} \]
Deep Momentum Nets Versus Deep Residual Nets

![Graph showing comparison between momentumnet and resnet](image-url)
CIFAR10 Classification

<table>
<thead>
<tr>
<th>Network</th>
<th># Params</th>
<th>ResNet (baseline)</th>
<th>MomentumNet (linear)</th>
<th>Improvement (linear/exponential)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PreResNet-110</td>
<td>1.1M</td>
<td>94.75 ± 0.14 (93.63 (He et al., 2016))</td>
<td>94.69 ± 0.12</td>
<td>-0.06</td>
</tr>
<tr>
<td>PreResNet-290</td>
<td>3.0M</td>
<td>94.95 ± 0.23</td>
<td>95.09 ± 0.11</td>
<td>0.14</td>
</tr>
<tr>
<td>PreResNet-470</td>
<td>4.9M</td>
<td>95.08 ± 0.10</td>
<td>95.09 ± 0.12</td>
<td>0.01</td>
</tr>
<tr>
<td>PreResNet-650</td>
<td>6.7M</td>
<td>95.13 ± 0.14</td>
<td>95.21 ± 0.13</td>
<td>0.08</td>
</tr>
<tr>
<td>PreResNet-830</td>
<td>8.6M</td>
<td>95.00 ± 0.23</td>
<td>95.31 ± 0.22</td>
<td>0.31</td>
</tr>
<tr>
<td>PreResNet-1001</td>
<td>10.3M</td>
<td>95.16 ± 0.19 (95.08 (He et al., 2016))</td>
<td>95.35 ± 0.15</td>
<td>0.19</td>
</tr>
</tbody>
</table>
ODE Models

\[ \dot{X} + \nabla f(X) = 0, \]

vs.

\[ \ddot{X} + \frac{3}{t} \dot{X} + \nabla f(X) = 0. \]

Su, Boyd, Candes, NIPS, 2014
Future Directions
Optimal restart scheduling in SRSGD.

How to generalize SRSGD to adaptive step size settings.

Neural architecture search with momentum units.
Microscopic Imaging (Cryo-EM)

**RELION**: Bayesian likelihood framework for cryo-EM structure determination:

\[
\underset{V_1, \ldots, K}{\text{arg max}} \log p(V_1, \ldots, K | X_1, \ldots, N) = \\
\underset{V_1, \ldots, K}{\text{arg max}} \sum_{i=1}^{N} \log \sum_{j=1}^{K} \frac{1}{K} \int p(X_i, \phi_i | V_j) d\phi_i + \log p(V_1, \ldots, K).
\]

The aim of the optimization is to find the 3D structures ($V_1$ to $V_K$) that best explain the observed images ($X_1$ to $X_N$) by marginalizing over class assignment ($j$) and the unknown pose variable ($\phi_i$) which describes a 3D rotation and a 2D translation for each single-particle image.

---

**Stochastic Gradient Descent!**

I. Scheduled Restart NAG Momentum
   I.1 Accelerate convergence
   I.2 Better generalization accuracy

II. Momentum in Neural Architecture Design
   II.1 Speed-up training
   II.2 Better generalization accuracy
   II.3 Mathematically mechanistic design

References: