AdaShift: Decorrelation and Convergence of Adaptive Learning Rate Methods

Zhiming Zhou*, Qingru Zhang*, Guansong Lu, Hongwei Wang, Weinan Zhang, Yong Yu

Apex Data & Knowledge Management Lab
Shanghai Jiao Tong University
Content

• Introduction

• Non-convergence behavior of Adam

• Theoretic Analysis

• AdaShift, the Algorithm proposed

• Experiment Results
Introduction

Adaptive Optimization Algorithm:

• General updating rule: \( \theta_{t+1} = \theta_t - \frac{\alpha_t}{\sqrt{v_t}} m_t \)

• Common choice of \( m_t \) and \( v_t \) is the exponential moving average of the gradients and squared gradients.

• Some state-of-art algorithms:
  • Adam, Adadelta, RMSProp, and Nadm.
  • Adam update rules:
    \[
    m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \\
    v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\
    \theta_t \leftarrow \theta_{t-1} - \alpha \cdot m_t / (\sqrt{v_t} + \epsilon)
    \]
Non-convergence Situations

“On the convergence of Adam and Beyond” pointed out two type of non-convergence problems for Adam:

• Sequential Counterexample:

\[ f_t(\theta) = \begin{cases} 
  C\theta, & \text{if } t \mod d = 1; \\
  -\theta, & \text{otherwise}, 
\end{cases} \]

• Stochastic Counterexample:

\[ f_t(\theta) = \begin{cases} 
  C\theta, & \text{with probability } p = \frac{1+\delta}{C+1}; \\
  -\theta, & \text{with probability } 1 - p = \frac{C-\delta}{C+1}, 
\end{cases} \]
Non-convergence Situations

Non-convergence Condition

• Sequential Counterexample:
  • For any fixed $\beta_1$ and $\beta_2$, $C$ need to satisfy:
    \[
    (1 - \beta_1)\beta_1^{C-1}C \leq 1 - \beta_1^{C-1}, \quad \beta_2^{(C-2)/2}C^2 \leq 1,
    \]
    \[
    \frac{3(1 - \beta_1)}{2\sqrt{1 - \beta_2}} \left(1 + \frac{\gamma(1 - \gamma^{C-1})}{1 - \gamma}\right) + \frac{\beta_1^{C/2-1}}{1 - \beta_1} < \frac{C}{3},
    \]

• Stochastic Counterexample:
  • For any fixed $\beta_1$ and $\beta_2$,
  • when $C$ is large enough (as a function of $\beta_1$, $\beta_2$, $\delta$),
  • the exception of update step will become non-negative

• Main Issue
  • Positive definiteness of $\Gamma_{t+1}$

\[
\Gamma_{t+1} = \left(\frac{\sqrt{V_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{V_t}}{\alpha_t}\right)
\]
Non-convergence Situations

Two solutions proposed by Reddi et al.

• AMSGrad

\[
v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2
\]

\[
\hat{v}_t = \max(\hat{v}_{t-1}, v_t)
\]

• Once a large gradient appears, it will maintain a very large \(v_t\)

• and slow down the training process

• AdamNC

• do not change the structure of Adam

• use an increasing schedule of \(\beta_2\), like \(\beta_{2t} = 1 - 1/t\)

• \(v_t\) equal to the average of all history gradients squared

• “long-term memory” but less flexibility

• slightly violate the positive definiteness of \(\Gamma_{t+1}\)
Non-convergence Condition

Stochastic Counterexample Experiments:

Conclusion:

• Both $\beta_1$ and $\beta_2$ influence the direction and speed of optimization

• Critical value of $C_t$, at which Adam gets into non-convergence, increases as $\beta_1$ and $\beta_2$ getting large.

• For any fixed $C$, as long as $\beta_1$ and $\beta_2$ large enough, non-convergence will disappear
The Cause of Non-Convergence

Unbalanced Step Size

• $v_t$ is positively correlated to the scale of gradient $g_t$
• It results in a small step size for a large gradient
• a large step size for a small gradient
• A common property of adaptive optimizer
Net Update Factor

To analyze the it we use a new perspective:

- Consider the effect of every gradients on the whole optimization process.

\[
net(g_t) \triangleq \sum_{i=t}^{\infty} \frac{\alpha_i}{\sqrt{v_i}} [(1 - \beta_1)\beta_1^{i-t} g_t] = k(g_t) \cdot g_t,
\]

where \( k(g_t) = \sum_{i=t}^{\infty} \frac{\alpha_i}{\sqrt{v_i}} (1 - \beta_1)\beta_1^{i-t} \)
Net Update Factor

Sequential Counterexample

• Limit of $v_t$
  \[
  \lim_{n \to \infty} v_{nd+i} = \frac{1 - \beta_2}{1 - \beta_2^d} (C^2 - 1) \beta_2^{i-1} + 1
  \]

• Limit of net update factor
  \[
  \lim_{n \to \infty} k(g_{nd+i}) = \sum_{t=nd+i}^{\infty} \frac{(1 - \beta_1) \beta_1^{t-nd-i}}{\sqrt{\frac{1 - \beta_2}{1 - \beta_2^d} (C^2 - 1) \beta_2^{(t-1) \mod d}}} + 1
  \]

• Conclusion: $k(C) < k(-1)$

Stochastic Counterexample

• For expectation of net update factor, $k(C) < k(-1)$

⇒ Unbalanced Step Size, combined with suitable $\beta_1$ and $\beta_2$, will cause the expectation of updates turn to non-negative
Decorrelation leads to convergence

Unbalanced step size is caused by the tight correlation between $v_t$ and $g_t$

Decorrelation will lead to convergence.

• [Theorem] If $v_t$ follows a fixed distribution and is independent of the current gradient $g_t$, then the expected net update factor for each gradient is identical.

Role of $v_t$

• $v_t$ reflects the gradient scale, and adjusts learning rate dynamically

• In AdaShift, current $v_t$ is independent with $g_t$, but the distribution of $v_t$ is close to $g_t$’s, and changes dynamically with $g_t$’s.
AdaShift, Decorrelation Variant

• Based on Adam, AdaShift adds two operations:
  
  *Temporal Shifting & Spatial Decorrelation*

• Algorithm:

\[
\text{Algorithm 2 Block-wise Temporal-Spatial Decorrelation}
\]

<table>
<thead>
<tr>
<th>Input: ( \theta_0, g_0, {f_t(\theta)}<em>{t=1}^T, {\alpha_t}</em>{t=1}^T ) and ( \beta_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: set ( v_0 = 0 )</td>
</tr>
<tr>
<td>2: for ( t = 1 ) to ( T ) do</td>
</tr>
<tr>
<td>3: ( g_t = \nabla f_t(\theta_t) )</td>
</tr>
<tr>
<td>4: for ( i = 1 ) to ( M ) do</td>
</tr>
<tr>
<td>5: ( v_t[i] = \beta_2 v_{t-1}[i] + (1 - \beta_2)\phi(g_{t-1}[i])^2 )</td>
</tr>
<tr>
<td>6: ( \theta_t[i] = \theta_{t-1}[i] - \alpha_t / \sqrt{v_t[i]} \cdot g_t[i] )</td>
</tr>
<tr>
<td>7: end for</td>
</tr>
<tr>
<td>8: end for</td>
</tr>
</tbody>
</table>
Intuitive Explanation

• Temporal Shifting:

\[
\begin{align*}
\ldots & \quad \ldots & \quad g_{t-n-2} & \quad g_{t-n-1} & \quad g_{t-n} & \quad g_{t-n+1} & \quad \ldots & \quad g_t \\
\phi & \quad \phi & \quad \phi \\
\ldots & \quad \ldots & \quad v_{t-2} & \quad v_{t-1} & \quad v_t & \quad m_t \\
\end{align*}
\]

at timestep \( t \)

• Spatial Decorrelation

For the gradient matrix of every layer, \( \varphi \) is a mapping function on it.

Matrix \( g_t \)

\[ \varphi(g_t) \]

Future Work:
The design of \( \varphi \)
Temporal Shifting

• Given the randomness of mini-batch, we assume that the mini-batch is independent of each other

• Thus, \( g_t \) is independent of each other in timeline

• The update rule for \( v_t \) now involves \( g_{t-1} \) (or \( g_{t-n} \)) instead of \( g_t \)

\[
v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_{t-1}^2
\]
Spatial Decorrelation, Layer-wise Adaptive Learning Rate

for $i = 1$ to $M$ do
    $v_t[i] = \beta_2 v_{t-1}[i] + (1 - \beta_2)[\phi(g_{t-1}[i])]^2$
    $\theta_t[i] = \theta_{t-1}[i] - \alpha_t / \sqrt{v_t[i] \cdot g_t[i]}$
end for

- no longer interpret $v_t$ as the second moment of $g_t$
- $v_t$ is a random variable, independent of $g_t$, while at the same time, reflects the overall gradient scale.
- Initialization methods somehow guarantee that the scale gradients in one layer are similar.
- Apply $\phi$ layer-wisely, outputs a shared adaptive learning rate scalar $v_t[i]$ ⇒ an adaptive learning rate SGD
- Adam sometimes does not generalize better than SGD, which might relate to the excessive learning rate adaptation in Adam
Conclusion:

• AdaShift will converge on the correct direction and converge at the fastest speed
Experiment

DenseNet with Cifar-10

DenseNet with Tiny-ImageNet
Conclusion:
AdaShift maintain a competitive performance with Adam in terms of both training speed and generalization.
Experiment

Training WGAN Discriminator

Neural Machine Translation BLEU
Extension

• Design of mapping function $\phi$
• Understanding on generalization between SGD and Adam
• Understanding on layer-wise optimization
• Unit-wise adaptive learning rate method
Q & A
Thanks for Listening!