Efficient Riemannian Optimization on the Stiefel Manifold via the Cayley Transform

Jun Li, Li Fuxin, Sinisa Todorovic
Oregon State University

ICLR 2020
Advantages of Orthonormality in Deep Learning

• Orthonormal matrices: $\{X \in \mathbb{R}^{n \times p}: X^TX = I\}, n \geq p$.

• Enforcing orthonormality on parameter matrices in deep learning:
  • Improves accuracy and empirical convergence rate (Bansal et al. 2018)
  • Stabilizes the distribution of neural activations in training (Huang et al. 2018)
  • Mitigates the vanishing and exploding-gradient problems (Zhou et al. 2006)
Prior Work

- **Soft Orthonormality -- Regularization:**
  - SO: \(\lambda||W^T W - I||_F^2\)
  - DSO: \(\lambda(||W^T W - I||_F^2 + ||WW^T - I||_F^2)\)
  - SRIP: \(\lambda \cdot \sigma(W^T W - I)\)

- **Limitation:** cannot enforce exact orthonormality

- **Hard Orthonormality -- Riemannian Optimization on the Stiefel manifold:**
  - Projection-based method: SVD
  - Retraction-based method: Closed form Cayley transform

- **Limitation:** computationally expensive
Our Contributions

- Improve computational efficiency of Riemannian optimization on the Stiefel manifold
  - Iterative Cayley transform that avoids the matrix inverse as a parameter update.
  - Implicit vector transport as a momentum update.

- Theoretical analysis of convergence of the proposed algorithm

- Faster convergence rate is empirically verified
**Preliminary**

**Manifold:** a topological space that locally resembles Euclidean space near each point

**Tangent Space:** a linear space that locally approximates the manifold
**Preliminary**

**Geodesic and Exp map:** a locally shortest curve on the manifold. Exponential map projects tangent vectors to geodesics. Exp map is a way to update parameters on a manifold.

**Parallel transport:** a way of transporting vectors along the geodesics while keep the norm. Parallel transport is a way to update momentum on a manifold.

Usually, exponential map and parallel transport are computationally expensive!
Usually, retraction and vector transport are computationally efficient.

**Retraction:** represent a smooth curve on a manifold

**Vector Transport:** an alternative way to move vectors along retractions on a manifold
Stiefel Manifold

**Stiefel manifold:**

A Riemannian manifold that consists of all $n \times p$ orthonormal matrices

$$\{ X \in \mathbb{R}^{n \times p} : X^T X = I \}$$

**Cayley Transform**

$$Y(\alpha) = (I - \alpha/2 W)^{-1}(I + \alpha/2 W)X$$

where $W$ is a skew-symmetric matrix

**Cayley Transform is a retraction on the Stiefel manifold**
Parameter Updates by Iterative Cayley Transform

Cayley Closed Form

\[ Y(\alpha) = (I - \frac{\alpha}{2} W)^{-1} (I + \frac{\alpha}{2} W) X \]

where \( W \) is a skew-symmetric matrix

Iterative Cayley Transform

\[ Y(\alpha) = X + \frac{\alpha}{2} W (X + Y(\alpha)) \]

Computationally efficient without matrix inversion! Numerically, two iterations are sufficient to achieve orthonormality.
Momentum Updates by the Implicit Vector Transport

Projection onto the tangent space is an implicit vector transport

$$\tau_{\eta_X}(\xi_X) = \pi_{T_{\eta_X}}(\xi_X)$$

By regarding the Stiefel manifold as an embedded submanifold of Euclidean space

where $$\rho_X(Z_1, Z_2) = tr(Z_1^T Z_2)$$

Implicit Momentum Updating

$$\alpha \tau_{M_k}(M_k) + \beta \nabla_M f(X_k)$$

$$= \alpha \pi_{T_{X_k}}(M_k) + \beta \pi_{T_{X_k}}(\nabla f(X_k))$$

$$= \pi_{T_{X_k}}(\alpha M_k + \beta \nabla f(X_k))$$

Computationally efficient with implicit momentum!
Proposed Algorithms

Algorithm 1 The Cayley SGD with Momentum

1: **Input:** learning rate \( lr \), momentum coefficient \( \beta \), \( \epsilon = 10^{-8} \), \( q = 0.5 \), \( s = 2 \).
2: Initialize \( X_1 \) as an orthonormal matrix; and \( M_1 = 0 \).
3: **for** \( k = 0 \) **to** \( T \) **do**
4: \( M_{k+1} \leftarrow \beta M_k - G(X_k) \), \( \text{Momentum updating} \)
5: \( \hat{W}_k \leftarrow M_{k+1} X_k^\top - \frac{1}{2} X_k (X_k^\top M_{k+1} X_k^\top) \), \( \text{Compute the auxiliary matrix} \)
6: \( W_k \leftarrow \hat{W}_k - \hat{W}_k \)
7: \( M_{k+1} \leftarrow W_k X_k \).
8: \( \alpha \leftarrow \min\{lr, 2q/(\|W_k\| + \epsilon)\} \)
9: Initialize \( Y^0 \leftarrow X + \alpha M_{k+1} \).
10: **for** \( i = 1 \) **to** \( s \) **do**
11: \( Y^i \leftarrow X_k + \frac{\alpha}{2} W_k (X_k + Y^{i-1}) \), \( \text{Parameter updating} \)
12: Update \( X_{k+1} \leftarrow Y^s \)
Proposed Algorithms

Algorithm 2 The Cayley ADAM

1: **Input:** learning rate $lr$, momentum coefficients $\beta_1$ and $\beta_2$, $\epsilon = 10^{-8}$, $q = 0.5$, $s = 2$.
2: Initialize $X_1$ as an orthonormal matrix. $M_1 = 0$, $v_1 = 1$
3: **for** $k = 0$ **to** $T$ **do**
4: $M_{k+1} \leftarrow \beta_1 M_k + (1 - \beta_1)G(X_k)$
5: $v_{k+1} \leftarrow \beta_2 v_k + (1 - \beta_2)\|G(X_k)\|^2$
6: $\hat{v}_{k+1} \leftarrow v_{k+1}/(1 - \beta_2^k)$
7: $r \leftarrow (1 - \beta_1^k)\sqrt{\hat{v}_{k+1}} + \epsilon$
8: $\hat{W}_k \leftarrow M_{k+1}X_k^T - \frac{1}{2}X_k(X_k^TM_{k+1}X_k^T)$
9: $W_k \leftarrow (\hat{W}_k - \hat{W}_k^T)/r$
10: $M_{k+1} \leftarrow rW_kX_k$
11: $\alpha \leftarrow \min\{lr, 2q/(\|W_k\| + \epsilon)\}$
12: Initialize $Y^0 \leftarrow X_k - \alpha M_{k+1}$
13: **for** $i = 1$ **to** $s$ **do**
14: $Y^i \leftarrow X_k - \frac{\alpha}{2} W(X_k + Y^{i-1})$
15: Update $X_{k+1} \leftarrow Y^s$
Convergence Analysis

**Assumption 1.** The gradient $\nabla f$ of the objective function $f$ is Lipschitz continuous

\[ \| \nabla f(X) - \nabla f(Y) \| \leq L \| X - Y \|, \quad \forall X, Y, \text{ where } L > 0 \text{ is a constant.} \]

**Theorem 1.** For $\alpha \in (0, \min\{1, \frac{2}{\|W\|}\})$, the iteration $Y^{i+1} = X + \frac{\alpha}{2} W (X + Y^i)$ is a contraction mapping and converges to the closed-form Cayley transform $Y(\alpha)$ given by Eq. 3. Specifically, at iteration $i$, $\| Y^i - Y(\alpha) \| = o(\alpha^{2+i})$.

Theorem 1 shows the iterative Cayley transform converges faster than other approximation algorithms, e.g., the Newton iterative.

**Theorem 2.** Given an objective function $f(X)$ that satisfies Assumption 1, let the Cayley SGD with momentum run for $t$ iterations with $G(X_k)$. For $\alpha = \min\{\frac{1-\beta}{L}, \frac{A}{\sqrt{t+1}}\}$, where $A$ is a positive constant, we have: $\min_{k=0, \ldots, t} E[\| \nabla \mathcal{M} f(X_k) \|^2] = o\left(\frac{1}{\sqrt{t+1}}\right) \to 0$, as $t \to \infty$.

Theorem 2 shows the proposed algorithm will eventually converge.
Training Loss Comparison in terms of Epoch

Training loss curves of different optimization algorithms for WRN-28-10. (a) Results on CIFAR10. (b) Results on CIFAR100. Both figures show that our Cayley SGD and Cayley ADAM achieve the top two fastest convergence rates in terms of epoch.
Training Loss Comparison in terms of Time

Training loss curves of different optimization algorithms for WRN-28-10. (a) Results on CIFAR10. (b) Results on CIFAR100. Both figures show that our Cayley SGD and Cayley ADAM achieve the top two fastest convergence rates in terms of time.
Comparison to SOTA

<table>
<thead>
<tr>
<th>Method</th>
<th>Error Rate(%)</th>
<th>Training time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CIFAR10</td>
<td>CIFAR100</td>
</tr>
<tr>
<td>Baselines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGD</td>
<td>3.89</td>
<td>18.66</td>
</tr>
<tr>
<td>ADAM</td>
<td>3.85</td>
<td>18.52</td>
</tr>
<tr>
<td>Soft orthonormality</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SO [3]</td>
<td>3.76</td>
<td>18.56</td>
</tr>
<tr>
<td>DSO [3]</td>
<td>3.86</td>
<td>18.21</td>
</tr>
<tr>
<td>SRIP [3]</td>
<td>3.60</td>
<td>18.19</td>
</tr>
<tr>
<td>Hard orthonormality</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMDSM [19]</td>
<td>3.73</td>
<td>18.61</td>
</tr>
<tr>
<td>DBN [20]</td>
<td>3.79</td>
<td>18.36</td>
</tr>
<tr>
<td>Polar [1]</td>
<td>3.75</td>
<td>18.50</td>
</tr>
<tr>
<td>QR [1]</td>
<td>3.75</td>
<td>18.65</td>
</tr>
<tr>
<td>Wen &amp; Yin [47]</td>
<td>3.82</td>
<td>18.70</td>
</tr>
<tr>
<td>Cayley closed form w/o momentum</td>
<td>3.80</td>
<td>18.68</td>
</tr>
<tr>
<td>Cayley SGD (Ours)</td>
<td>3.66</td>
<td>18.26</td>
</tr>
<tr>
<td>Cayley ADAM (Ours)</td>
<td>3.57</td>
<td>18.10</td>
</tr>
</tbody>
</table>

Error rate and training time per epoch comparison to baselines with WRN-28-10 on CIFAR10 and CIFAR100. All experiments are performed on one TITAN Xp GPU.
## Experiments for RNN

<table>
<thead>
<tr>
<th>Model</th>
<th>Hidden Size</th>
<th>Closed-Form</th>
<th></th>
<th>Cayley SGD</th>
<th></th>
<th>Cayley ADAM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Acc(%)</td>
<td>Time(s)</td>
<td>Acc(%)</td>
<td>Time(s)</td>
<td>Acc(%)</td>
<td>Time(s)</td>
</tr>
<tr>
<td>Full-uRNN</td>
<td>116</td>
<td>92.8</td>
<td>2.10</td>
<td>92.6</td>
<td>1.42</td>
<td>92.7</td>
<td>1.50</td>
</tr>
<tr>
<td>Full-uRNN</td>
<td>512</td>
<td>96.9</td>
<td>2.44</td>
<td>96.7</td>
<td>1.67</td>
<td><strong>96.9</strong></td>
<td>1.74</td>
</tr>
</tbody>
</table>

Table 4: Pixel-by-pixel MNIST accuracy and training time per iteration of the closed-form Cayley Transform, Cayley SGD, and Cayley ADAM for Full-uRNNs (Wisdom et al., 2016). All experiments are performed on one TITAN Xp GPU.
Table 5: Checking unitariness by computing the error $\|K^H K - I\|_F$ for varying numbers of iterations in the iterative Cayley transform and the closed-form Cayley transform.
Conclusion

• We specified a scalable method to enforce the exact orthonormal constraints on parameters of deep learning networks.

• SGD and ADAM are generalized to Cayley SGD with momentum and Cayley ADAM on the Stiefel manifold.

• Theoretical analysis of convergence of the two algorithms is provided.

• Experiments show that both algorithms achieve comparable performance and faster convergence over the baseline SGD and ADAM.

• Both Cayley SGD with momentum and Cayley ADAM take less runtime per epoch than all existing hard orthonormal methods and soft orthonormal methods, and can be applied to non-square parameter matrices.
• Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep cnns? (NeurIPS 2018)
• Lei Huang, Xianglong Liu, Bo Lang, Adams Wei Yu, Yongliang Wang, and Bo Li. Orthogonal weight normalization: Solution to optimization over multiple dependent Stiefel manifolds in deep neural networks (AAAI 2018)
• Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints
• P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Optimization algorithms on matrix manifolds.
• P-A Absil and Jerome Malick. Projection-like retractions on matrix manifolds.
• Gary Becigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. (ICLR 2019)
Thank You

ACKNOWLEDGEMENT:
NSF grant IIS-1911232, DARPA XAI Award N66001-17-2-4029, AFRL STTR AF18B-T002.