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: $\{X \in R^{n \times \mathrm{p}} : X^T X = I\}, n \geq p$.
- 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: $\lambda||W^TW - I||_F^2$
  - DSO: $\lambda(||W^TW - I||_F^2 + ||WW^T - I||_F^2)$
  - SRIP: $\lambda \cdot \sigma(W^TW - I)$
  - 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

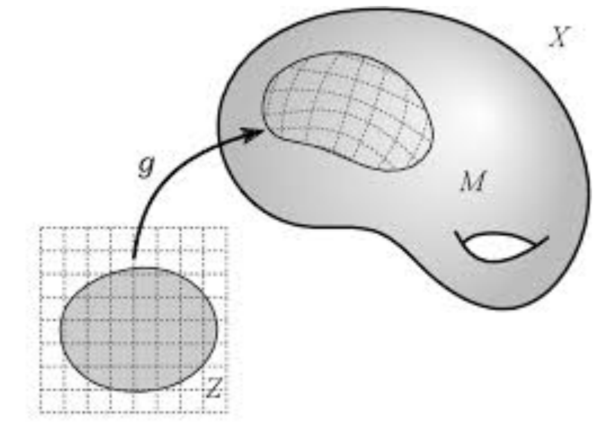
Our Contributions

- 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.

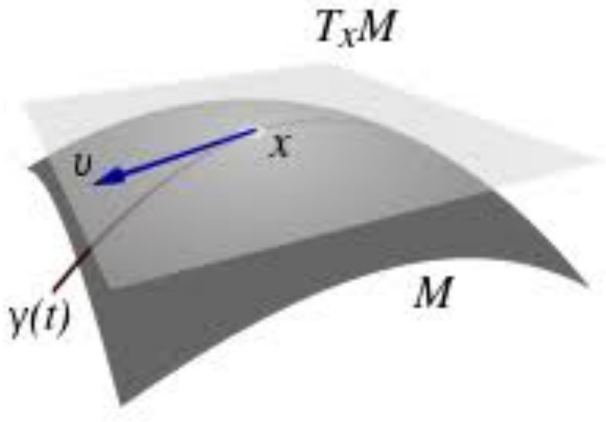
- Theoretical analysis of convergence of the proposed algorithm

- Faster convergence rate is empirically verified

Preliminary

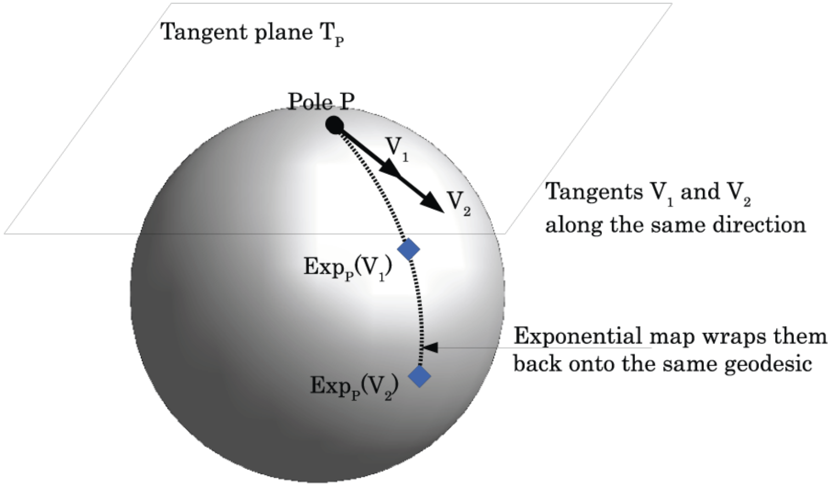


Manifold: a topological space that locally resembles Euclidean space near each point

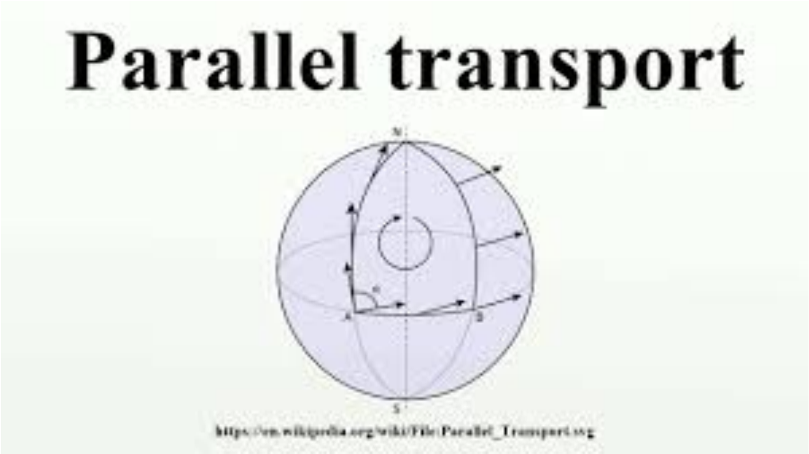


Tangent Space: a linear space that locally approximates the manifold

Preliminary



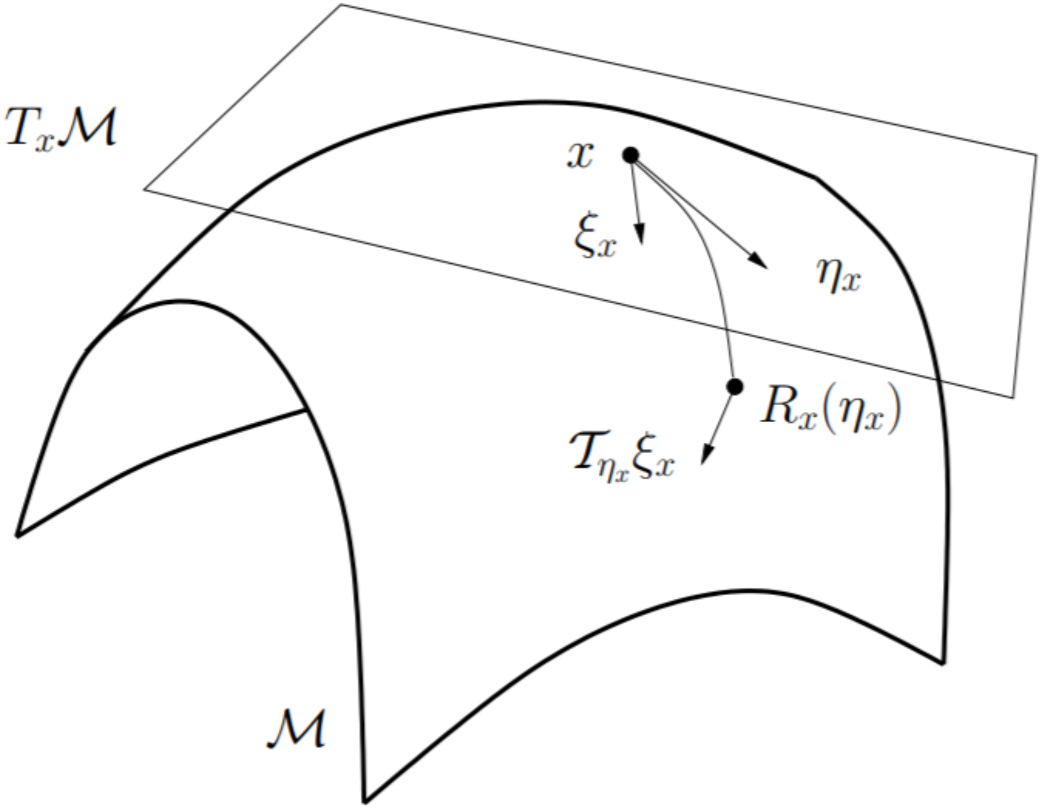
Geodesic and Exp map: a locally shortest curve on the manifold. Exponential map projects tangent vectors to geodesics. Exp map is a way to update parameters on a manifold.



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!

Preliminary

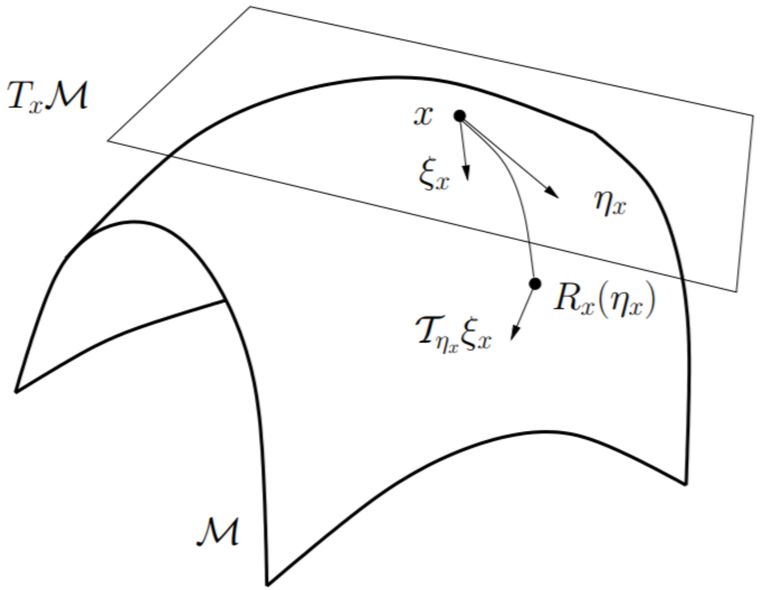


Retraction: represent a smooth curve on a manifold

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

Usually, retraction and vector transport are computationally efficient.

Stiefel Manifold



Stiefel manifold:

a Riemannian manifold that consists of all n × p orthonormal matrices

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

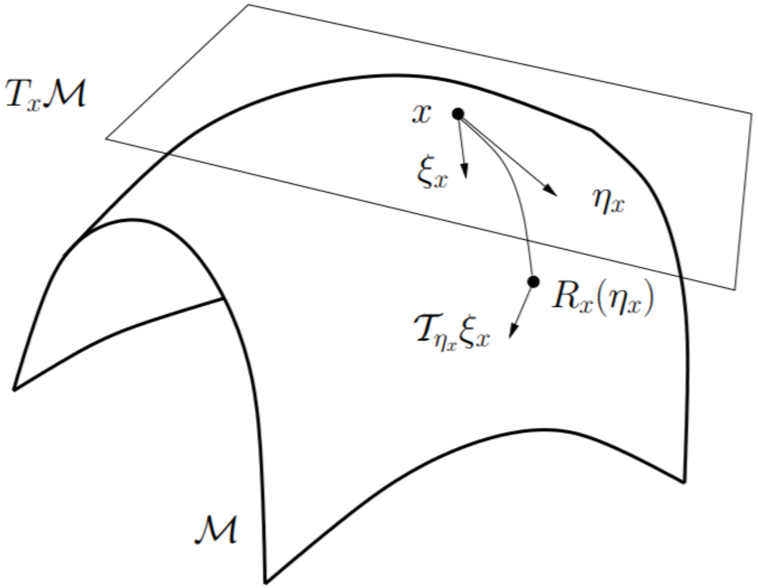
Cayley Transform

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

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

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

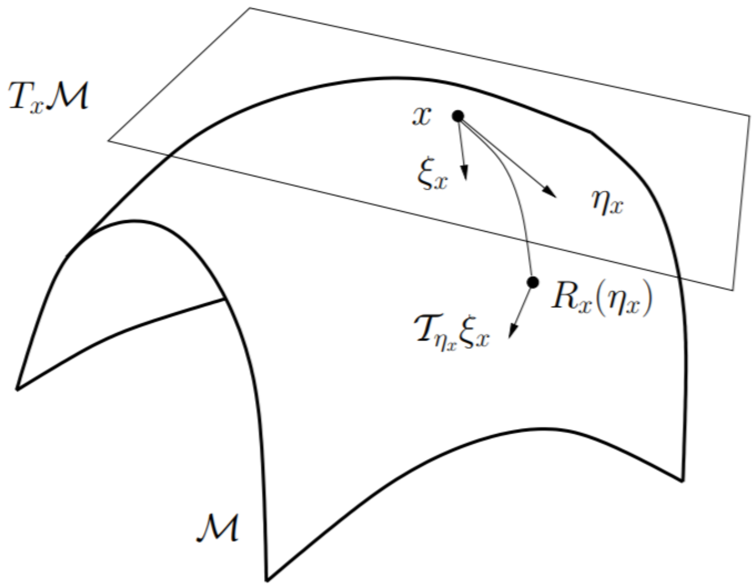
where W is a skew-symmetric matrix

Iterative Cayley Transform

$$Y(\alpha) = X + \frac{\alpha}{2}W\left(X + Y(\alpha)\right)$$

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

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

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

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

Implicit Momentum Updating

Projection of gradient: Inherent in the Cayley transform

$$\begin{aligned} &\alpha\tau_{M_k}(M_k) + \beta\nabla_{\mathcal{M}} f(X_k) \\ &= \alpha\pi_{T_{X_k}}(M_k) + \beta\pi_{T_{X_k}}(\nabla f(X_k)) \\ &= \pi_{T_{X_k}}(\alpha M_k + \beta\nabla f(X_k)) \end{aligned}$$

Computationally efficient with implicit momentum!

Proposed Algorithms

Algorithm 1 The Cayley SGD with Momentum

Input: learning rate lr, momentum coefficient β, $\epsilon{=}10^{-8}$, $q = 0.5$, $s = 2$.
Initialize X_1 as an orthonormal matrix; and $M_1 = 0$
for $k = 0$ **to** T **do**
  $M_{k+1} \leftarrow \beta M_k - \mathcal{G}(X_k)$, ⟵ Momentum updating ▷ Update the momentum
  $\hat{W}_k \leftarrow M_{k+1} X_k^\top - \frac{1}{2} X_k (X_k^\top M_{k+1} X_k^\top)$ ▷ *Compute the auxiliary matrix*
  $W_k \leftarrow \hat{W}_k - \hat{W}_k^\top$
  $M_{k+1} \leftarrow W_k X_k$. ▷ *Project momentum onto the tangent space*
  $\alpha \leftarrow \min\{lr, 2q/(\|W_k\| + \epsilon)\}$ ▷ *Select adaptive learning rate for contraction mapping*
  Initialize $Y^0 \leftarrow X + \alpha M_{k+1}$ ▷ *Iterative estimation of the Cayley Transform*
  for $i = 1$ **to** s **do**
    $Y^i \leftarrow X_k + \frac{\alpha}{2} W_k (X_k + Y^{i-1})$ ⟵ Parameter updating
  Update $X_{k+1} \leftarrow Y^s$

Proposed Algorithms

Algorithm 2 The Cayley ADAM

Input: learning rate lr, momentum coefficients β_1 and β_2, $\epsilon = 10^{-8}$, $q = 0.5$, $s = 2$.

Initialize X_1 as an orthonormal matrix. $M_1 = 0$, $v_1 = 1$

for $k = 0$ **to** T **do**

$M_{k+1} \leftarrow \beta_1 M_k + (1-\beta_1)\mathcal{G}(X_k)$ ▷ *Estimate biased momentum*

Momentum updating

$v_{k+1} \leftarrow \beta_2 v_k + (1-\beta_2)\|\mathcal{G}(X_k)\|^2$

$\hat{v}_{k+1} \leftarrow v_{k+1}/(1-\beta_2^k)$ ▷ *Update biased second raw moment estimate*

$r \leftarrow (1-\beta_1^k)\sqrt{\hat{v_{k+1}} + \epsilon}$ ▷ *Estimate biased-corrected ratio*

Manifold-wise adaptive learning rate

$\hat{W}_k \leftarrow M_{k+1}X_k^\top - \frac{1}{2}X_k(X_k^\top M_{k+1} X_k^\top)$ ▷ *Compute the auxiliary skew-symmetric matrix*

$W_k \leftarrow (\hat{W}_k - \hat{W}_k^\top)/r$

$M_{k+1} \leftarrow rW_kX_k$ ▷ *Project momentum onto the tangent space*

$\alpha \leftarrow \min\{lr, 2q/(\|W_k\| + \epsilon)\}$ ▷ *Select adaptive learning rate for contraction mapping*

Initialize $Y^0 \leftarrow X_k - \alpha M_{k+1}$ ▷ *Iterative estimation of the Cayley Transform*

for $i = 1$ **to** s **do**

$Y^i \leftarrow X_k - \frac{\alpha}{2}W(X_k + Y^{i-1})$

Parameter updating

Update $X_{k+1} \leftarrow Y^s$

Convergence Analysis

Assumption 1. *The gradient ∇f of the objective function f is Lipschitz continuous*

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

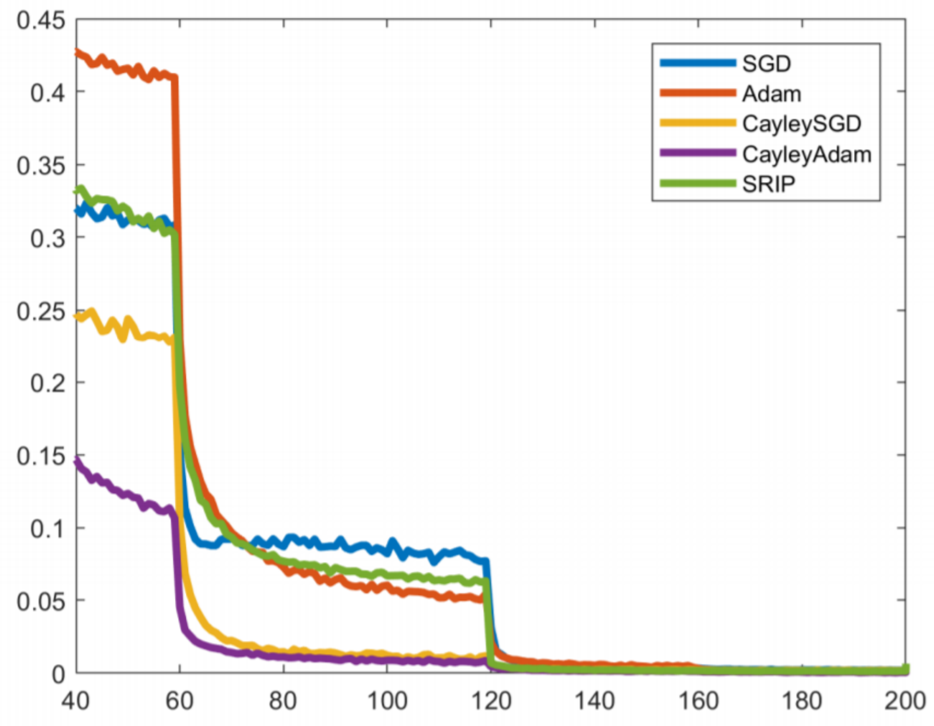
Theorem 1. *For $\alpha \in (0, \min\{1, \frac{2}{\|W\|}\})$, the iteration $Y^{i+1} = X + \frac{\alpha}{2}W\left(X + Y^i\right)$ is a contraction mapping and converges to the closed-form Cayley transform $Y(\alpha)$ given by Eq. 3. Specifically, at iteration i, $||Y^i - Y(\alpha)|| = o(\alpha^{2+i})$.*

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

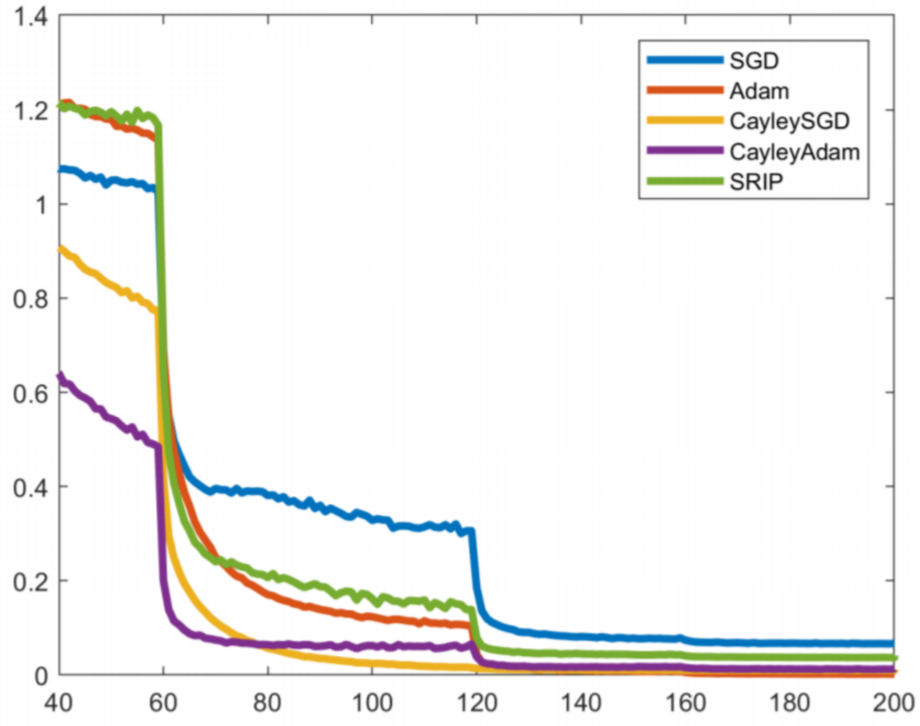
Theorem 2. *Given an objective function $f(X)$ that satisfies Assumption 1, let the Cayley SGD with momentum run for t iterations with $\mathcal{G}(X_k)$. For $\alpha = \min\{\frac{1-\beta}{L}, \frac{A}{\sqrt{t+1}}\}$, where A is a positive constant, we have: $\min_{k=0,\cdots,t} E[\|\nabla_{\mathcal{M}}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

Theorem 2 shows the proposed algorithm will eventually converge.

Training Loss Comparison in terms of Epoch



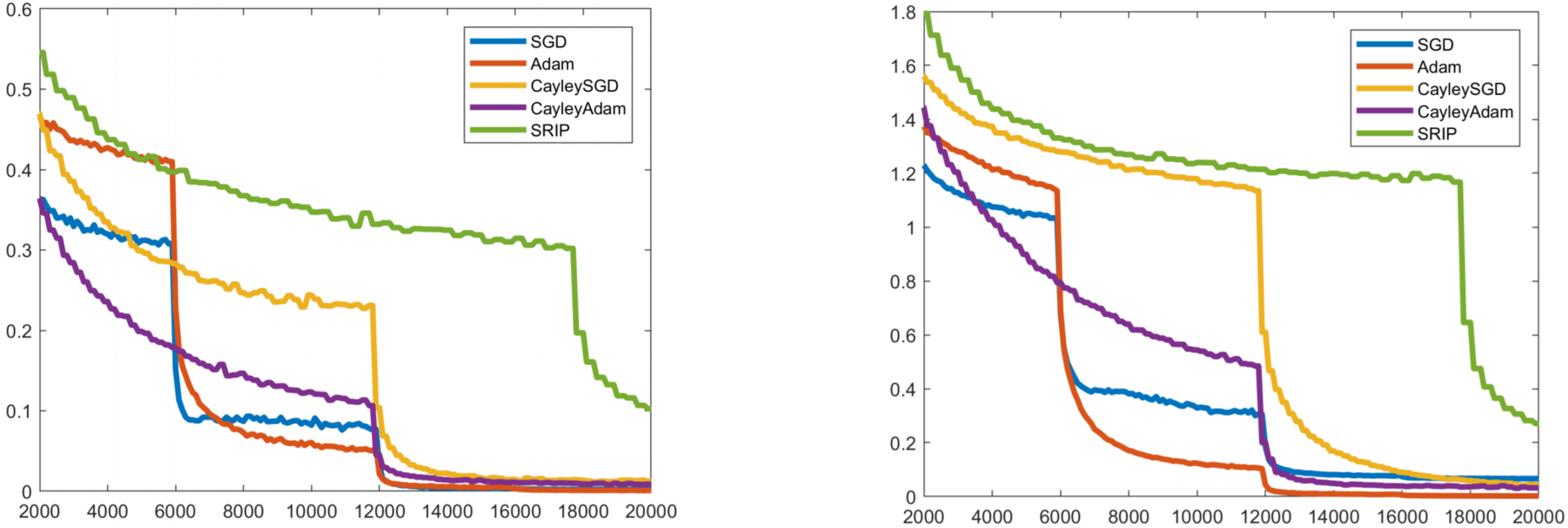
(a) CIFAR10



(b) CIFAR100

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



(a) CIFAR10 (b) CIFAR100

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

	Method	Error Rate(%) CIFAR10	CIFAR100	Training time(s)
Baselines	SGD	3.89	18.66	102.5
	ADAM	3.85	18.52	115.2
Soft orthonormality	SO [3]	3.76	18.56	297.3
	DSO [3]	3.86	18.21	311.0
	SRIP [3]	3.60	18.19	321.8
Hard orthonormality	OMDSM [19]	3.73	18.61	943.6
	DBN [20]	3.79	18.36	889.4
	Polar [1]	3.75	18.50	976.5
	QR [1]	3.75	18.65	469.3
	Wen&Yin [47]	3.82	18.70	305.8
	Cayley closed form w/o momentum	3.80	18.68	1071.5
	Cayley SGD (**Ours**)	3.66	18.26	218.7
	Cayley ADAM (**Ours**)	3.57	18.10	224.4

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

Model	Hidden Size	Closed-Form		Cayley SGD		Cayley ADAM	
		Acc(%)	Time(s)	Acc(%)	Time(s)	Acc(%)	Time(s)
Full-uRNN	116	92.8	2.10	92.6	1.42	92.7	1.50
Full-uRNN	512	96.9	2.44	96.7	1.67	**96.9**	1.74

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.

Check Unitariness

Hidden Size	s=0	s=1	s=2	s=3	s=4	Closed-form
n=116	3.231e-3	2.852e-4	7.384e-6	7.353e-6	7.338e-6	8.273e-5
n=512	6.787e-3	5.557e-4	2.562e-5	2.547e-5	2.544e-5	3.845e-5

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.

Reference

- 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)
- Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks (ICML 2016)
- 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 Jer´ ome Malick. Projection-like retractions on matrix manifolds.

- Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks (ICML 2016)

- 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.

OSU
Oregon State
UNIVERSITY
College of Engineering