Fast and accurate optimization on the orthogonal manifold without retractions

Pierre Ablin
CNRS-PSL University-Université Paris-Dauphine
pierre.ablin@dauphine.psl.eu

We let $f : \mathbb{R}^{p\times p} \to \mathbb{R}$ a smooth function, and consider the problem of minimizing f over the orthogonal manifold $\mathcal{O}_p = \{X \in \mathbb{R}^{p\times} |\ X^\top X = I_p\}$. We study iterative algorithms that produce a sequence of iterates X_k that should converge to the solution of the problem. In order to find X_{k+1}, Riemannian gradient descent [1] first computes the Riemannian gradient G_k, i.e. the projection of $\nabla f(X_k)$ in the tangent space at X_k, which is the linear space $T_{X_k} = \{AX_k |\ A^\top = -A\}$. Simple computations give $G_k = \mathrm{Skew}(\nabla f(X_k)X_k^\top)X_k$. This algorithm then uses a retraction to move in the opposite direction while staying on the manifold. For instance, the classical exponential retraction gives $X_{k+1} = \exp(-\eta\mathrm{Skew}(\nabla f(X_k)X_k^\top))X_k$, with $\eta > 0$ a step size: it is straightforward to check that if X_k is orthogonal, then X_{k+1} is still orthogonal, and that as η gets small, we have $X_{k+1} \simeq X_k - \eta G_k$. Unfortunately, the numerical computation of retractions on the orthogonal manifold always involves some expensive linear algebra operation, such as matrix inversion, exponential or square-root. These operations quickly become expensive as the dimension p grows.

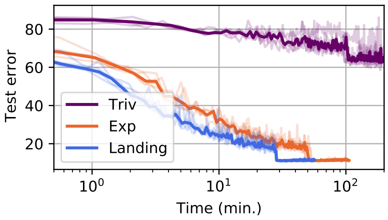

Figure 1: Learning curves for a deep residual network with orthogonal weights on CIFAR10

To bypass this limitation, we propose the landing algorithm which does not use retractions. Letting $\mathcal{N}(X) = \frac{1}{4}\|X^\top X - I_p\|_F^2$ the "distance" to the manifold, we define the landing field as

$$\Lambda(X) = \mathrm{Skew}(\nabla f(X)X^\top)X + \lambda\nabla\mathcal{N}(X),$$

and the landing algorithm simply iterates $X_{k+1} = X_k - \eta\Lambda(X_k)$. The algorithm is not constrained to stay on the manifold but the term $\nabla\mathcal{N}(X)$ progressively attracts it towards the manifold.

One iteration of the landing algorithm only involves matrix multiplications, which makes it cheap compared to its retraction counterparts, especially on modern hardware like GPU's. Fig 2 illustrates the computational cost of the landing field compared to most classical retractions. Theoretically, we show that the algorithm converges with the usual rate for a non-convex problem: with small enough step-size η, we get $\sup_{k\geq K}\mathcal{N}(X_k) = O(\frac{1}{K})$ and $\sup_{k\geq K}\|G_k\|^2 = O(\frac{1}{K})$, showing that the algorithm reaches stationary points of the optimization problem at a $1/\sqrt{K}$ rate, just like Riemannian gradient descent [2]. Numerical experiments demonstrate the promises of our approach in settings where computing retractions is very costly, such as training of deep neural networks with orthogonal weights. Fig. 1 displays the test error of a deep residual network with orthogonal weights trained on the CIFAR 10 dataset: the landing method is the fastest.

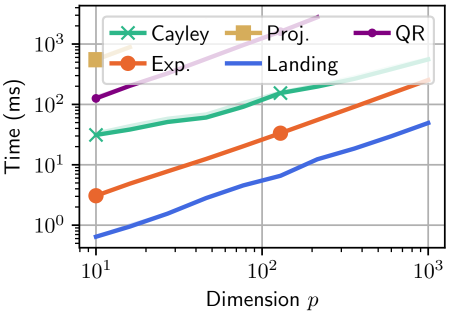

Figure 2: Time required to compute 500 retractions when A and X are of size $p \times p$, on a GPU.

Joint work with: Gabriel Peyré (CNRS - PSL University - ENS)

References

[1] Pierre-Antoine Absil, Robert Mahony and Rodolphe Sepulchre. Optimization algorithms on matrix manifolds. *Princeton University Press*, 2009.

[2] Nicolas Boumal, Pierre-Antoine Absil, and Coralia Cartis. Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis* 39.1 (2019): 1-33.