Numerically Stable Optimization on Hyperbolic Space

Seunghyuk Cho[1] Dongjun Yu[2]

[1]Graduate School of Computer Science, [2]Graduate School of Artificial Intelligence

Midway Presentation, May 2022

Table of Contents

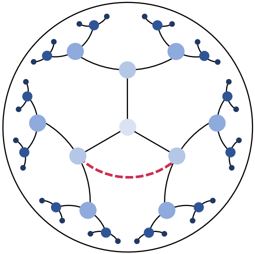
1. Hyperbolic space
2. Riemannian SGD
3. Our idea

Table of Contents

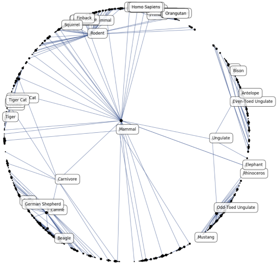
1. Hyperbolic space
2. Riemannian SGD
3. Our idea

Hyperbolic space

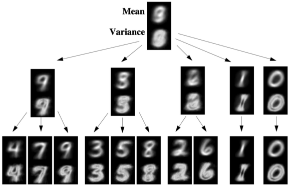
Hyperbolic space has shown outstanding performance on learning embeddings of hierarchical data.



Hyperbolic space



Word relationships[1]



MNIST[2]

---

[1] (Maximillian Nickel and Douwe Kiela. "Poincaré embeddings for learning hierarchical representations". In: *Advances in neural information processing systems* 30 [2017])

[2] (Ruslan Salakhutdinov, Joshua Tenenbaum, and Antonio Torralba. "One-Shot Learning with a Hierarchical Nonparametric Bayesian Model". In: *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. Vol. 27. Proceedings of Machine Learning Research. 2012, pp. 195–206)

Hyperbolic space

To learn the parameters on the hyperbolic space, we need to solve the following constrained optimization problem:

$$\min_{x \in \mathcal{L}^n} f(x),$$

where $\mathcal{L}^n$ denotes the n-dimensional hyperbolic space.

Table of Contents

1. Hyperbolic space
2. Riemannian SGD
3. Our idea

Riemannian SGD

Stochastic gradient descent on Riemannian manifolds

S. Bonnabel *

Riemannian Manifold

Riemannian manifold $(\mathcal{M}, g)$ is a pair of a manifold and a metric tensor.

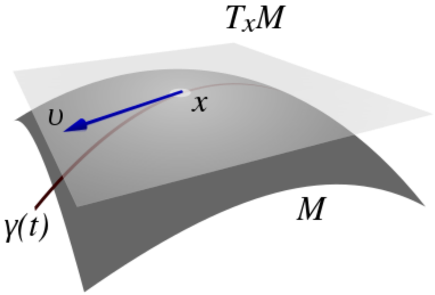
- Manifold $\mathcal{M}$: set of points.
- Metric tensor g: induce basic geometric operations, i.e. distance.

Examples

Hyperbolic space is the unique, complete, simply connected Riemannian manifold with constant negative sectional curvature.

Tangent Space

A tangent space $T_x\mathcal{M}$ is a set of tangent vectors which are tangent to the manifold $\mathcal{M}$ at x.

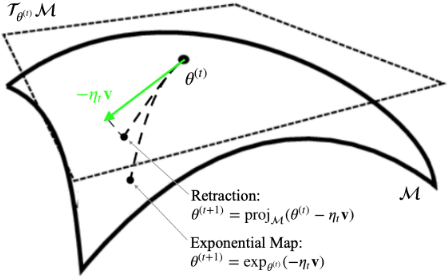


Riemannian SGD

We can update the parameters on the Riemannian manifold as:

$$x_{t+1} = \exp_{x_t}(-\eta_t H(x_t)),$$

where $H = g^{-1}\nabla f$ is the Riemannian gradient ($H(x) \in T_x\mathcal{M}$).



Riemannian SGD

Theorem 1. *Consider the algorithm* (2) *on a connected Riemannian manifold* $\mathcal{M}$ *with injectivity radius uniformly bounded from below by* $I > 0$. *Assume the sequence of step sizes* $(\gamma_t)_{t \geq 0}$ *satisfy the standard condition* (4). *Suppose there exists a compact set* K *such that* $w_t \in K$ *for all* $t \geq 0$. *We also suppose that the gradient is bounded on* K, *i.e. there exists* $A > 0$ *such that for all* $w \in K$ *and* $z \in \mathcal{Z}$ *we have* $\|H(z, w)\| \leq A$. *Then* $C(w_t)$ *converges a.s. and* $\nabla C(w_t) \to 0$ *a.s.*

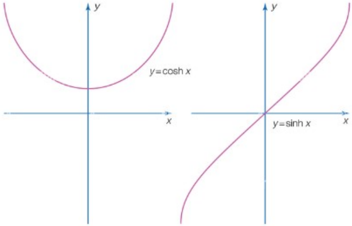
$$\sum \gamma_t^2 < \infty \quad \text{and} \quad \sum \gamma_t = +\infty \tag{4}$$

Table of Contents

1. Hyperbolic space
2. Riemannian SGD
3. Our idea

Previous work

Previous work argue that the operations of the hyperbolic space are numerically unstable due to the hyperbolic functions.[1][2]



[1] Weize Chen et al. "Fully Hyperbolic Neural Networks". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022.

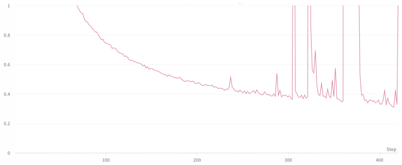
[2] Maximillian Nickel and Douwe Kiela. "Learning continuous hierarchies in the lorentz model of hyperbolic geometry". In: *International Conference on Machine Learning*. 2018, pp. 3779–3788.

Our idea

We empirically show that the Riemannian SGD is numerically unstable due to the exponential map.

range	[-1, 1]	[-10, 10]	[-100, 100]	[-1000, 1000]
Lorentz (float)	3.29E-14	0.00E+00	Nan	Nan
Lorentz (double)	7.70E-34	0.00E+00	2.03E-28	Nan
Poincare (float)	0.00E+00	5.05E+01	6.46E+03	8.06E+05
Poincare (double)	1.25E-32	4.97E+00	6.15E+03	9.98E+05

Error rate of $\exp^{-1}(\exp(\cdot))$



Unstable loss curve

Our idea

Q) Then, why did the author have to use this method?
A) The constraint set in the paper is too complex and so is the projection or the regularizer.

Since our constraint set $\mathcal{L}^n$ is relatively simple, we can use some simple methods.

Methods

Indirect Method

$y_{t+1} = y_t - \eta_t \nabla f(\exp_{0_{\mathcal{L}}}(\text{concat}(0, y_t)))$, where $y \in \mathbb{R}^n$

Projected Method

$x_{t+1} = \text{proj}(x_t - \eta_t \nabla f(x_t))$

Landing Algorithm

$x_{t+1} = x_t - \eta_t \nabla h(x_t)$, where $h(x) = f(x) + \lambda R(x)$

Thank you for listening!