Hyperbolic vs Euclidean Embeddings in Few-Shot Learning: Two Sides of the Same Coin

Supplementary Material

A. The hyperboloid model

Consider the upper-sheet of a d-dimensional hyperboloid $H_k^d \subset \mathbb{R}^{d,1}$ as defined in (2). At each of its points $\mathbf{x} \in H_k^d$, we have the tangent space as $T_{\mathbf{x}}H_k^d = \{\mathbf{v} \in \mathbb{R}^{d,1} : \langle \mathbf{x}, \mathbf{v} \rangle_L = 0\}$. For $\mathbf{v} \in T_{\mathbf{x}}H_k^d$, there is a unique geodesic $\gamma_{\mathbf{v}} : \mathbb{R} \to H_k^d$ such that $\gamma'_{\mathbf{v}}(0) = \mathbf{v}$. The exponential map $\mathrm{Exp}^H : TH_k^d \to H_k^d$ is defined as $\mathrm{Exp}_{\mathbf{x}}^H(\mathbf{v}) := \gamma_{\mathbf{v}}(1)$,

$$\mathrm{Exp}_x^H(\mathbf{v}) := \gamma_{\mathbf{v}}(1) = \cosh\left(\|\mathbf{v}\|_L\sqrt{-k}\right)\mathbf{x} + \frac{\sinh\left(\|\mathbf{v}\|_L\sqrt{-k}\right)\mathbf{v}}{\|\mathbf{v}\|_L\sqrt{-k}}, \tag{23}$$

where $\|\mathbf{v}\|_L = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle_L}$, with $\langle \cdot, \cdot \rangle_L$ the Lorentz pseudometric (1). Given two points $\mathbf{x}, \mathbf{y} \in H_k^d$, the hyperbolic distance $d_{H_k^d}(\mathbf{x}, \mathbf{y})$ is obtained by integrating the velocity of the geodesic between them,

$$d_{H_k^d}(\mathbf{x}, \mathbf{y}) := \frac{1}{\sqrt{-k}}\mathrm{acosh}(k\langle \mathbf{x}, \mathbf{y} \rangle_L). \tag{24}$$

We can verify that the hyperboloid H_k^d and the Poincaré ball P_k^d are isometric models of d-dimensional hyperbolic space with curvature k *i.e.*, for $\mathbf{x}, \mathbf{y} \in H_k^d$

$$d_{H_k^d}(\mathbf{x}, \mathbf{y}) = d_{P_k^d}\left(\Pi(\mathbf{x}), \Pi(\mathbf{y})\right), \tag{25}$$

where Π is the stereographic projection defined in (4). A comparison of geometries with constant positive, zero and negative curvature is presented in Table 6. Note that $\langle \cdot, \cdot \rangle_E$ denotes the Euclidean inner product (dot product).

Manifold	Curvature	Geodesic $d(\mathbf{x}, \mathbf{y})$
Euclidean $\mathbb{R}^d$	$k = 0$	$\sqrt{\langle \mathbf{x} - \mathbf{y}, \mathbf{x} - \mathbf{y} \rangle_E}$
Spherical $S_k^d \subset \mathbb{R}^{d+1}$	$k > 0$	$\frac{1}{\sqrt{k}}\mathrm{acos}(k\langle \mathbf{x}, \mathbf{y} \rangle_E)$
Hyperbolic $H_k^d \subset \mathbb{R}^{d,1}$	$k < 0$	$\frac{1}{\sqrt{-k}}\mathrm{acosh}(k\langle \mathbf{x}, \mathbf{y} \rangle_L)$

Table 6. Overview of the different isotropic geometries.

Hyperbolic distance for fixed-radius embeddings Consider $\mathbf{x}$ and $\mathbf{y}$ in P_k^d such that $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = r$ and $\angle(\mathbf{x}, \mathbf{y}) = \alpha$. The hyperbolic distance between $\mathbf{x}$ and $\mathbf{y}$ can be computed using (25). Recall that $\lambda(\mathbf{u}) = 2/(1 + k\|\mathbf{u}\|_2^2)$. We have then

$$\begin{aligned}\left\langle \Pi^{-1}(\mathbf{x}), \Pi^{-1}(\mathbf{y}) \right\rangle_L &= \lambda(\mathbf{x})\lambda(\mathbf{y})\left\langle \mathbf{x}, \mathbf{y} \right\rangle_E + \frac{1}{k}(\lambda(\mathbf{x}) - 1)(\lambda(\mathbf{y}) - 1) \\ &= \frac{4}{(1 + kr^2)^2}r^2\cos(\alpha) + \frac{1}{k}\left(\frac{1 - kr^2}{1 + kr^2}\right)^2.\end{aligned} \tag{26}$$

Plugging this in (24) yields expression (21),

$$d_{P_k^d}(\mathbf{x}, \mathbf{y}) = d_{H_k^d}(\Pi^{-1}(\mathbf{x}), \Pi^{-1}(\mathbf{y})) = \frac{1}{\sqrt{-k}}\mathrm{acosh}\left(\frac{4kr^2}{(1 + kr^2)^2}\cos(\alpha) + \left(\frac{1 - kr^2}{1 + kr^2}\right)^2\right). \tag{27}$$

B. Models

Backbone Similarly to [13], the convolutional backbone used in all experiments consists of a sequence of 4 convolutional blocks, each of which composed of 3×3 2D Convolutions with 64 filters and stride 1, 2D Batch Normalization, ReLU activation and 2D MaxPool. The 4th block has as many filters as dimensions in the output manifold.

Scheduler A StepLR scheduler was used to train all models. In the CUB dataset, the initial learning rate of 10^{-3} is decayed by a factor of 0.8 every 40 epochs both in the 1s5w and the 5s5w few-shot settings. In the MiniImageNet dataset, the initial learning rate of 5×10^{-3} is decayed by a factor of 0.5 every 80 epochs in the 1s5w setting (trained as 1s30w), and by 0.5 every 60 epochs in the 5s5w scenario (trained as 5s20w).

Image transformations In the case of the CUB_200_2011, we crop the images according to the bounding boxes provided in the dataset before other image transformations. The data augmentations performed during training were: 1) Zero padding along the smallest dimension to produce a square image; 2) Random crop resized to $84 \times 84 \times 3$; 3) Image jitter with 0.4 brightness, 0.4 contrast and 0.4 hue; 4) Random horizontal flip; 5) Normalization. At test time, the images were zero padded, resized to 84×84 and normalized.