Hyperbolic Neural Networks

Octavian Ganea, Gary Bécigneul, Thomas Hofmann

Presented by Tyler Hattori

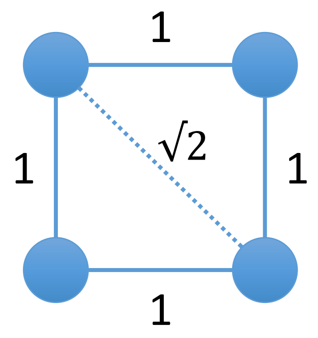
Presentation Outline

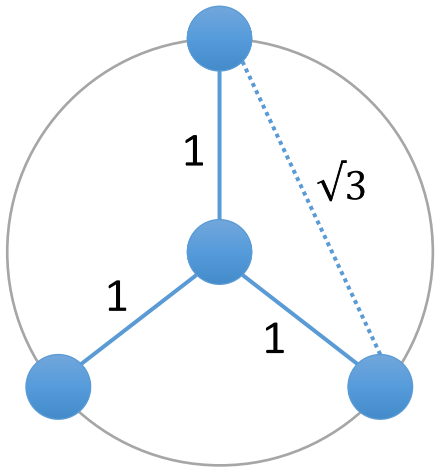
- Paper Overview
- Mathematical Background
- Hyperbolic Neural Networks
  - MLR, FFNN, RNN, GRU
- Results

Paper Overview

Motivation

- Embedding large classes of graphs is difficult with Euclidean geometry
- Hyperbolic geometry is well-suited for handling tree-like data
- However, we lack generalizations of basic operations and objects for hyperbolic geometry, so hyperbolic NN layers are unrealizable

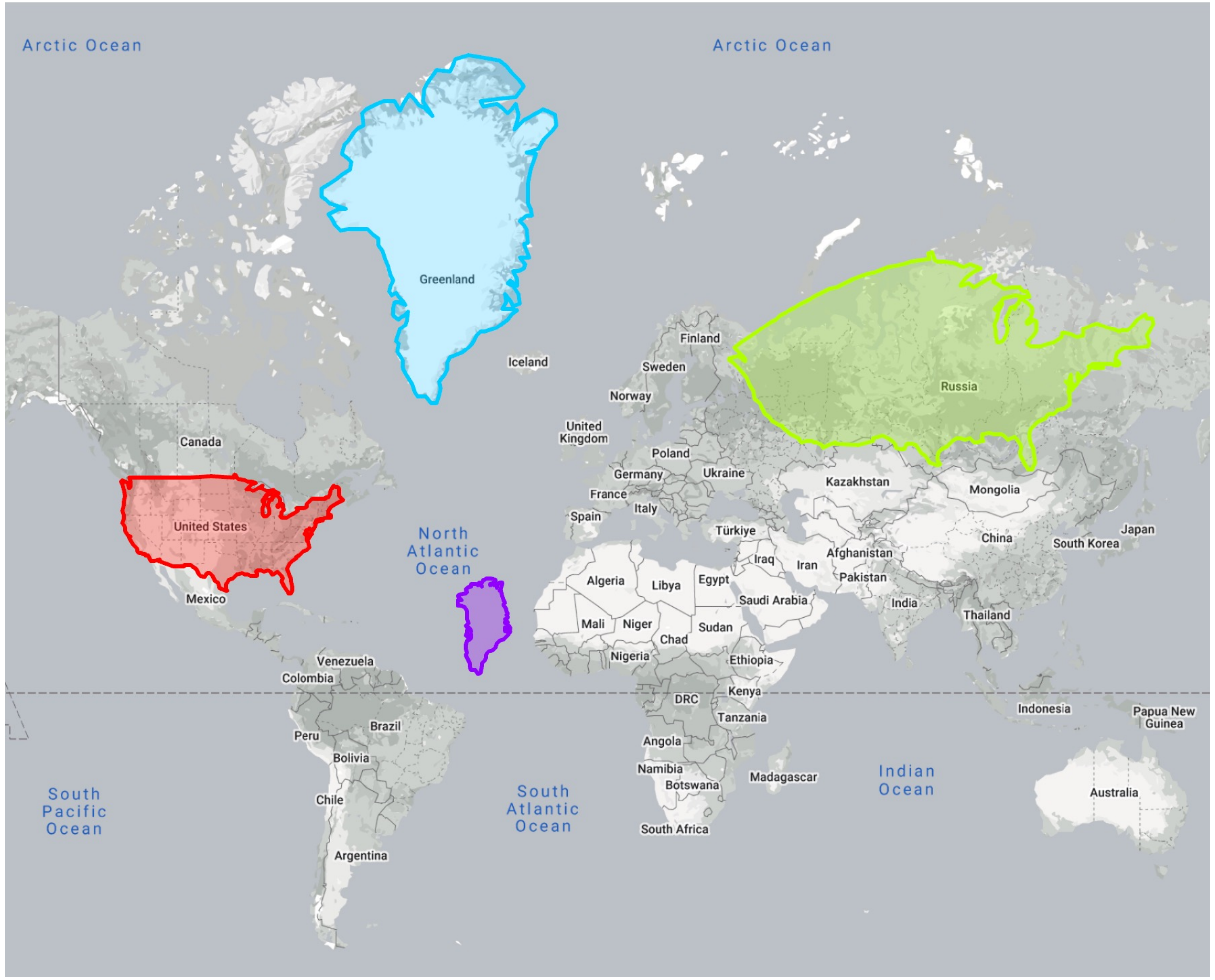




Paper Overview

Motivation

- Embedding large classes of graphs is difficult with Euclidean geometry
- Hyperbolic geometry is well-suited for handling tree-like data
- However, we lack generalizations of basic operations and objects for hyperbolic geometry, so hyperbolic NN layers are unrealizable

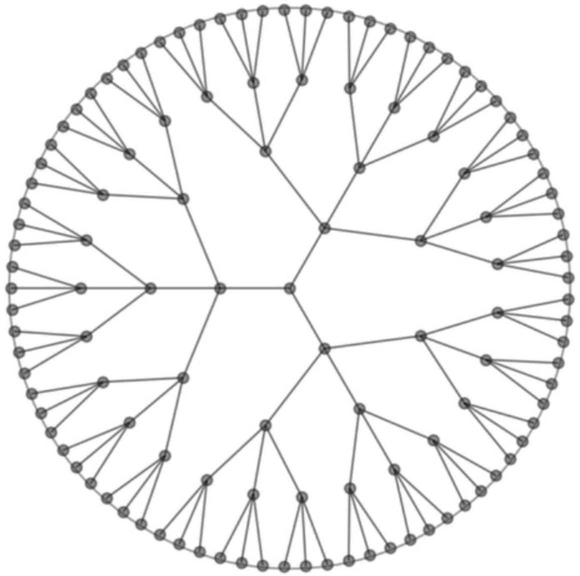


https://www.thetruesize.com

Paper Overview

Motivation

- Embedding large classes of graphs is difficult with Euclidean geometry
- Hyperbolic geometry is well-suited for handling tree-like data
- However, we lack generalizations of basic operations and objects for hyperbolic geometry, so hyperbolic NN layers are unrealizable



Tree structure

Hyperbolic geometry

https://www.youtube.com/watch?v=zQo_S3yNa2w

Paper Overview

Motivation

- Embedding large classes of graphs is difficult with Euclidean geometry
- Hyperbolic geometry is well-suited for handling tree-like data
- However, we lack generalizations of basic operations and objects for hyperbolic geometry, so hyperbolic NN layers are unrealizable

Operations

Vector addition
Matrix-vector multiplication
Translation
Inner product

Objects

Distances
Geodesics
Parallel Transport

?

Hyperbolic Geometry

Paper Overview

Contributions

- Developed a framework that parametrizes basic operations and objects in hyperbolic geometry with respect to a constant curvature
- Applied these generalizations to existing Euclidean NN models (MLR, FFNN, RNN, GRU)
- Showed improvement for sentence embedding tasks

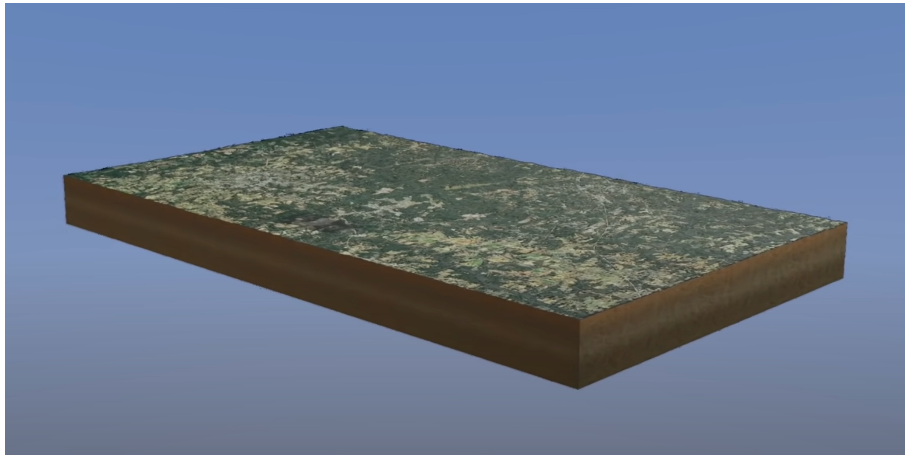
Background

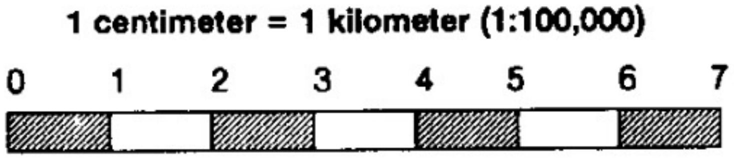
Riemannian geometry basic definitions

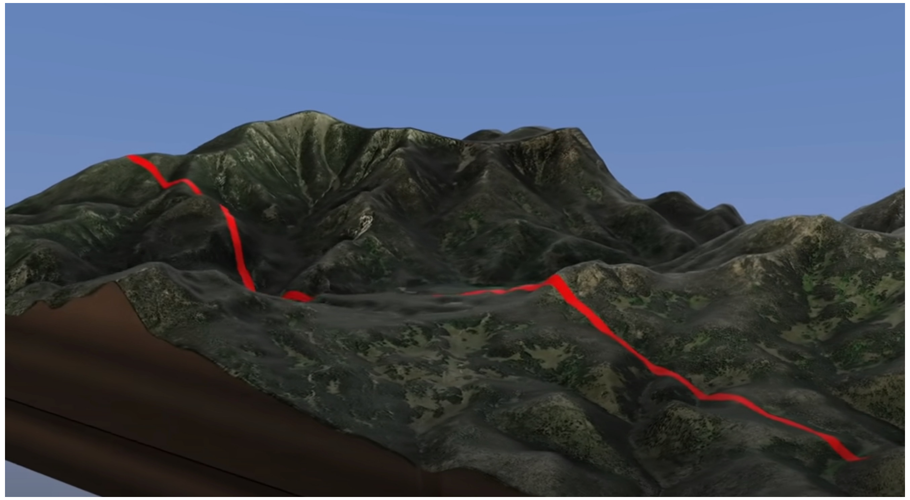
- An $n - dim$ **manifold** M is a space that can be locally approximated by $\mathbb{R}^n$
- The **tangent space** T_xM at $x \in M$ is a first order linear approximation of M around x
- The **exponential map** $\exp_x(v)$ projects a vector v on T_xM to a point on M
- A **geodesic** γ is the shortest smooth path between two points x and y on M. It is parametrized using $\exp_x(v)$ such that $\gamma(0) = x$, $\dot{\gamma}(0) = v$, $\gamma(1) \mapsto \exp_x(v) = y$
- A **Riemannian metric tensor** g_x defines the collection of positive-definite inner products taken on T_xM varying smoothly with x. g_x is used to define many geometric notions of M
- For example, this metric realizes the **distance function** $d(x, y)$ by integrating along the length of the geodesic from 0 to 1. $d(x, y) = \int_0^1 \sqrt{g_{\gamma(t)}(v, v)} dt$

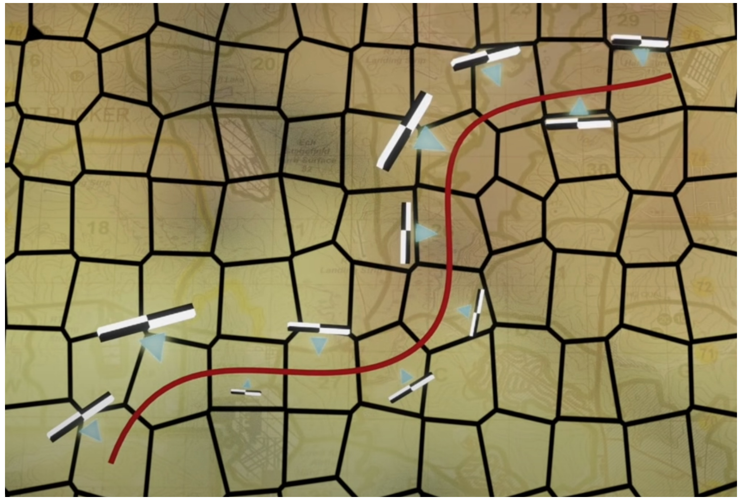
Background

Riemannian metric tensor intuition









https://www.youtube.com/watch?v=Hf-BxbtCg_A

Background

Riemannian geometry basic definitions

- The **parallel transport** $P_{x \to y}: T_x M \to T_y M$ takes a vector on $T_x M$ and moves it along the geodesic γ to output a vector on $T_y M$
- **Conformal** metrics are angle-preserving such that $\tilde{g}_x = \lambda_x^2 g_x$ for all x

Background

Hyperbolic Geometry

- The n-dim hyperbolic space is the unique, simply connected, n-dim, geodesically complete **Riemannian manifold** with constant curvature -1
  - Riemannian manifold → a manifold defined with a Riemannian metric tensor to describe its geometry
  - Unique → there is a unique geodesic between any two points on the manifold
  - Simply connected → no holes
  - Constant curvature -1 → intuitive opposite of a sphere
  - Geodesically complete → from any point, can follow a straight line in any direction

Hyperbolic Geometry

- The **Poincaré ball model** $(\mathbb{D}_c^n, g^c)$ is a **Riemannian manifold** $\mathbb{D}_c^n = \{x \in \mathbb{R}^n : c \parallel x \parallel < 1\}$ with

$$g_x^{\mathbb{D}} = \lambda_x^2 g^E, \quad \lambda_x := \frac{2}{1 - \|x\|^2}$$

- The Poincaré metric tensor $g_x^{\mathbb{D}}$ is conformal to the Euclidean one. From this, we find that the distance and angle functions on the manifold are

$$d_{\mathbb{D}}(x, y) = \cosh^{-1}\left(1 + 2\frac{\|x-y\|^2}{(1-\|x\|^2)(1-\|y\|^2)}\right) \qquad \cos(\angle(u,v)) = \frac{g_x^{\mathbb{D}}(u,v)}{\sqrt{g_x^{\mathbb{D}}(u,u)}\sqrt{g_x^{\mathbb{D}}(v,v)}} = \frac{\langle u, v\rangle}{\|u\|\|v\|}$$

- The Poincaré model exists inside the unit circle. All straight lines in the model are perpendicular to the unit circle.

- If $c \to 0$, the manifold becomes Euclidean $(\mathbb{D}_c^n \to \mathbb{R}^n)$. This paper defines a framework for hyperbolic operations parametrized on c so that the Euclidean and hyperbolic geometries can be continuously deformed into one another.

Hyperbolic Geometry

- Vector spaces form an algebraic setting for Euclidean space. Likewise, **gyrovector spaces** form an algebraic setting for hyperbolic geometries. Just as vector spaces in Euclidean space operate on groups, gyrovector spaces operate on **gyrogroups**. A gyrogroup $(G, \oplus)$ satisfies the following axioms

$$
\begin{aligned}
&(\mathrm{G1}) \quad 0 \oplus a = a \\
&(\mathrm{G2}) \quad \ominus a \oplus a = 0 \\
&(\mathrm{G3}) \quad a \oplus (b \oplus c) = (a \oplus b) \oplus \mathrm{gyr}[a, b] c \\
&(\mathrm{G4}) \quad \mathrm{gyr}[a, b] \in \mathrm{Aut}(G, \oplus) \\
&(\mathrm{G5}) \quad \mathrm{gyr}[a, b] = \mathrm{gyr}[a \oplus b, b]
\end{aligned}
$$

$$
\mathrm{gyr}[a, b] = \frac{a \oplus b}{b \oplus a} = \frac{1 + a\bar{b}}{1 + \bar{a}b}
$$

- The operation $\mathrm{gyr}[a, b]\colon G \times G \to \mathrm{Aut}(G, \oplus)$ can be thought of as a rotation
- Instead of associativity and communitivity, we have gyroassociativity and gyrocommunitivity

Hyperbolic Geometry

Basic Operations

- The **Mobius gyrovector space** is used in this paper to define basic operations on the Poincaré manifold.
- Mobius **addition** of elements on $\mathbb{D}_c^n$:

$$x \oplus_c y := \frac{(1 + 2c\langle x, y\rangle + c\|y\|^2)x + (1 - c\|x\|^2)y}{1 + 2c\langle x, y\rangle + c^2\|x\|^2\|y\|^2}$$

- Mobius **scalar multiplication**

$$r \otimes_c x := (1/\sqrt{c}) \tanh(r \tanh^{-1}(\sqrt{c}\|x\|)) \frac{x}{\|x\|}\ , \qquad r \otimes_c \mathbf{0} := \mathbf{0}$$

- Hyperbolic **distance**

$$d_c(x, y) = (2/\sqrt{c}) \tanh^{-1}\left(\sqrt{c}\| - x \oplus_c y\|\right)$$

- Considering $c = 0$ for the operations above yields intuitive Euclidean definitions

https://www.geogebra.org/m/GhaSJw3t

Hyperbolic Geometry

Paper Contributions

- **How can we connect gyrovector spaces with the Riemannian geometry of the Poincaré model?**
- We can describe geodesics on the Poincaré model using Mobius operations
- We can then define the exponential map and parallel transport in terms of Mobius operations
- This leads to a much easier representation of Mobius scalar multiplication
- Finally, this representation leads to an easy way of defining matrix vector multiplication on the Poincaré model, which allows us to formulate NN layers in hyperbolic space

Hyperbolic Geometry

- Define geodesic using Mobius operations

$$\gamma_{x \to y}(t) := x \oplus_c (-x \oplus_c y) \otimes_c t \qquad \gamma_{x \to y} : \mathbb{R} \to \mathbb{D}_c^n \text{ s.t. } \gamma_{x \to y}(0) = x \text{ and } \gamma_{x \to y}(1) = y$$

- Reparametrize to unit speed using the distance function. This is the unit speed geodesic from point x in direction v

$$\gamma_{x,v}(t) = x \oplus_c \left(\tanh\left(\sqrt{c}\frac{t}{2}\right) \frac{v}{\sqrt{c}\|v\|} \right) \qquad \gamma_{x,v} : \mathbb{R} \to \mathbb{D}^n \; s.t. \; \gamma_{x,v}(0) = x \; and \; \dot{\gamma}_{x,v}(0) = v$$

- Exponential and logarithmic (inverse) maps

$$\exp_x^c(v) = x \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_x^c \|v\|}{2}\right) \frac{v}{\sqrt{c}\|v\|} \right), \; \log_x^c(y) = \frac{2}{\sqrt{c}\lambda_x^c} \tanh^{-1}(\sqrt{c}\|-x \oplus_c y\|) \frac{-x \oplus_c y}{\|-x \oplus_c y\|}$$

- Simplified mappings for starting at the point $x \in 0$

$$\exp_{\mathbf{0}}^c(v) = \tanh(\sqrt{c}\|v\|) \frac{v}{\sqrt{c}\|v\|}, \; \log_{\mathbf{0}}^c(y) = \tanh^{-1}(\sqrt{c}\|y\|) \frac{y}{\sqrt{c}\|y\|}$$

Hyperbolic Geometry

- Define Mobius scalar multiplication using these mappings

$$r \otimes_c x = \exp_{\mathbf{0}}^c(r \log_{\mathbf{0}}^c(x)), \quad \forall r \in \mathbb{R}, x \in \mathbb{D}_c^n$$

- This means we can project a point on the manifold to the tangent space at 0, scale by r in Euclidean space, and project back to the manifold to achieve Pointcaré scalar multiplication

- We can define the Poincaré geodesic in a similar way

$$\gamma_{x \to y}(t) = x \oplus_c (-x \oplus_c y) \otimes_c t = \exp_x^c(t \log_x^c(y)), \quad t \in [0, 1]$$

- The same goes for the parallel transport

$$P_{\mathbf{0} \to x}^c(v) = \log_x^c(x \oplus_c \exp_{\mathbf{0}}^c(v)) = \frac{\lambda_{\mathbf{0}}^c}{\lambda_x^c} v.$$

Hyperbolic Neural Networks

Feed Forward (FFNN)

- For a mapping function $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, the Mobius map is

$$f^{\otimes_c}(x) := \exp_{\mathbf{0}}^c(f(\log_{\mathbf{0}}^c(x)))$$

- The Mobius matrix-vector multiplication is

$$M^{\otimes_c}(x) = (1/\sqrt{c}) \tanh\left(\frac{\|Mx\|}{\|x\|} \tanh^{-1}(\sqrt{c}\|x\|)\right) \frac{Mx}{\|Mx\|}$$

- A hyperbolic pointwise non-linearity $\varphi: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is also simply defined by its Mobius version $\varphi^{\otimes c}$

- Hyperbolic translation by a bias $b \in \mathbb{D}_c^n$ employs the Mobius parallel transport

$$x \oplus_c b = \exp_x^c(P_{\mathbf{0}\rightarrow x}^c(\log_{\mathbf{0}}^c(b))) = \exp_x^c\left(\frac{\lambda_{\mathbf{0}}^c}{\lambda_x^c} \log_{\mathbf{0}}^c(b)\right)$$

Hyperbolic Neural Networks

Softmax (MLR)

- Define a hyperplane on the Poincaré model as

$$\tilde{H}_{a,p}^{c} := \{x \in \mathbb{D}_c^n : \langle \log_p^c(x), a \rangle_p = 0\} = \exp_p^c(\{a\}^{\perp}) \qquad p \in \mathbb{D}_c^n,\ a \in T_p\mathbb{D}_c^n \setminus \{\mathbf{0}\}$$

- Define the shortest distance between a given point and the hyperplane as

$$d_c(x, \tilde{H}_{a,p}^{c}) := \inf_{w \in \tilde{H}_{a,p}^{c}} d_c(x, w) = \frac{1}{\sqrt{c}} \sinh^{-1}\left(\frac{2\sqrt{c}|\langle -p \oplus_c x, a \rangle|}{(1 - c\| - p \oplus_c x\|^2)\|a\|}\right)$$

- Hyperbolic MLR formula

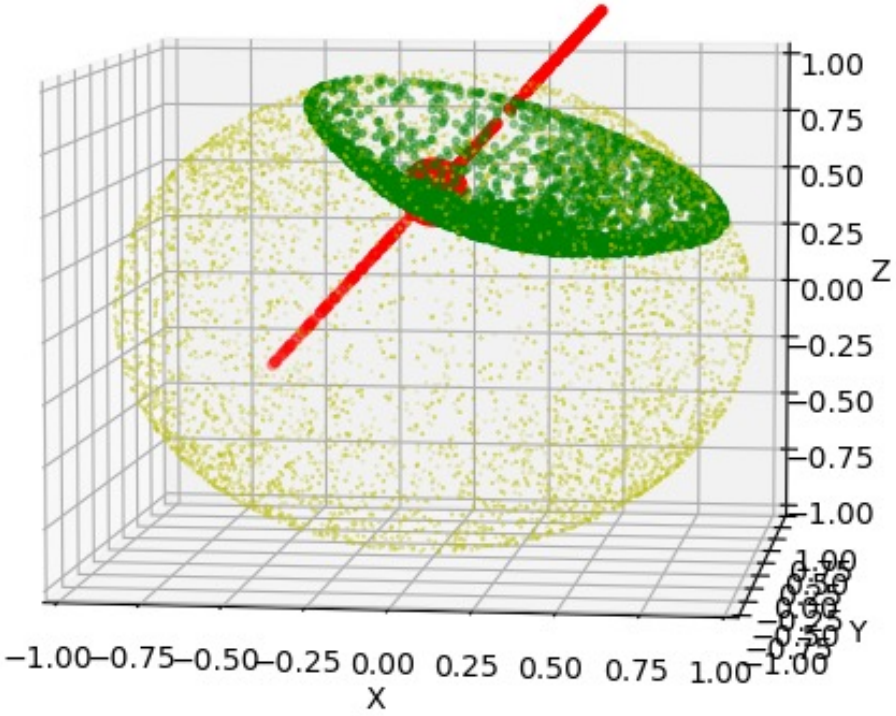
$$p(y = k|x) \propto \exp\left(\frac{\lambda_{p_k}^c \|a_k\|}{\sqrt{c}} \sinh^{-1}\left(\frac{2\sqrt{c}\langle -p_k \oplus_c x, a_k \rangle}{(1 - c\| - p_k \oplus_c x\|^2)\|a_k\|}\right)\right), \quad \forall x \in \mathbb{D}_c^n$$

- When $c = 0$, $p(y = k|x)$ becomes the Euclidean Softmax function

$$p(y = k|x) = \exp(\langle -p_k + x, a_k \rangle_0)$$

Hyperbolic Neural Networks

Softmax (MLR)



Hyperbolic hyperplane used for MLR code example

Hyperbolic Neural Networks

Recurrent (RNN)

- A simple RNN is defined by

$$h_{t+1} = \varphi(Wh_t + Ux_t + b)$$

where φ is some point non-linearity

- After defining the relevant Mobius functions, it is now easy to define a Hyperbolic RNN

$$h_{t+1} = \varphi^{\otimes_c}(W \otimes_c h_t \oplus_c U \otimes_c x_t \oplus_c b), \quad h_t \in \mathbb{D}_c^n,\ x_t \in \mathbb{D}_c^d$$

Hyperbolic Neural Networks

Gated Recurrent (GRU)

- A simple GRU is defined by

$$r_t = \sigma(W^r h_{t-1} + U^r x_t + b^r), \qquad z_t = \sigma(W^z h_{t-1} + U^z x_t + b^z),$$
$$\tilde{h}_t = \varphi(W(r_t \odot h_{t-1}) + U x_t + b), \quad h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

  where φ is some point wise product

- Ganea et al. adapts the pointwise product $r_t \odot h_{t-1}$ to $\text{diag}(r_t) \otimes_c h_{t-1}$

- They also adapt the update-gate mechanism and independently apply the hyperbolic model

- Therefore, the Hyperbolic GRU is defined by

$$r_t = \sigma \log_{\mathbf{0}}^c(W^r \otimes_c h_{t-1} \oplus_c U^r \otimes_c x_t \oplus_c b^r)$$
$$\tilde{h}_t = \varphi^{\otimes_c}((W \text{diag}(r_t)) \otimes_c h_{t-1} \oplus_c U \otimes_c x_t \oplus b)$$
$$h_t = h_{t-1} \oplus_c \text{diag}(z_t) \otimes_c (-h_{t-1} \oplus_c \tilde{h}_t)$$

Experiment 1

Sentence Entailment Binary Classification

- Goal: given two sentences, (1) a premise and (2) a hypothesis, have the model predict if the second sentence is inferred from the first one
- Architecture: (1) embed the two sentences using an RNN or GRU, (2) feed the embeddings and their distances to a FFNN, (3) apply binary MLR with cross-entropy loss on top

	SNLI	PREFIX-10%	PREFIX-30%	PREFIX-50%
FULLY EUCLIDEAN RNN	**79.34** %	89.62 %	81.71 %	72.10 %
HYPERBOLIC RNN+FFNN, EUCL MLR	**79.18** %	96.36 %	**87.83** %	**76.50** %
FULLY HYPERBOLIC RNN	78.21 %	**96.91** %	87.25 %	62.94 %
FULLY EUCLIDEAN GRU	**81.52** %	95.96 %	86.47 %	75.04 %
HYPERBOLIC GRU+FFNN, EUCL MLR	79.76 %	**97.36** %	**88.47** %	**76.87** %
FULLY HYPERBOLIC GRU	**81.19** %	**97.14** %	**88.26** %	**76.44** %

Experiment 2

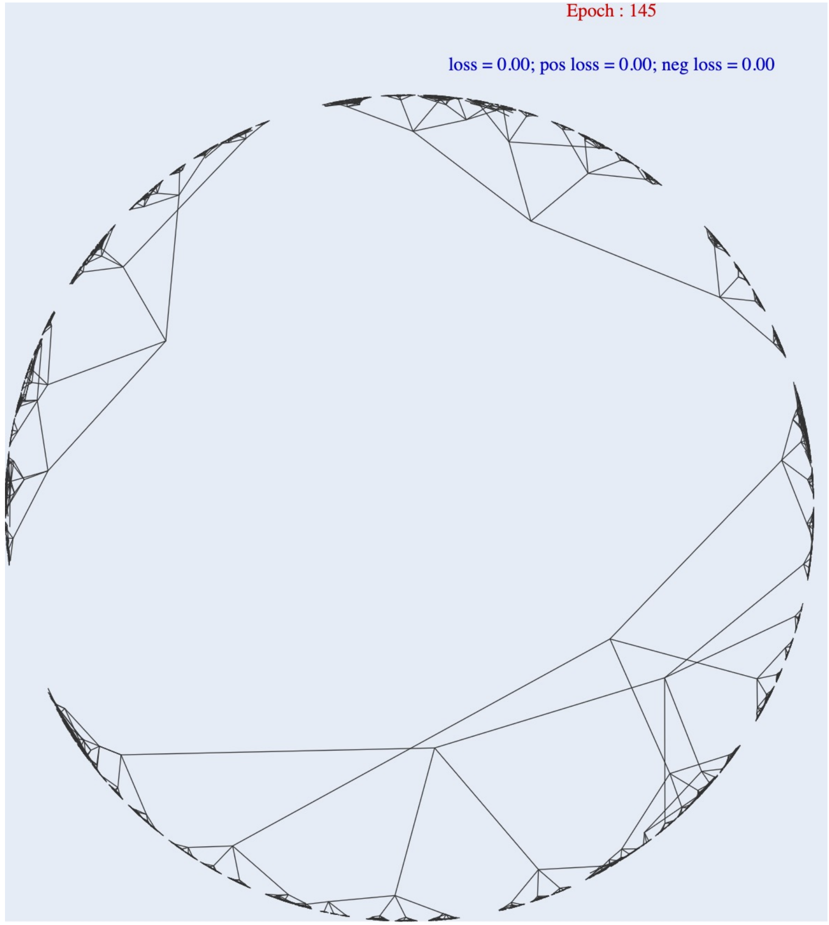
Noun Binary Classification

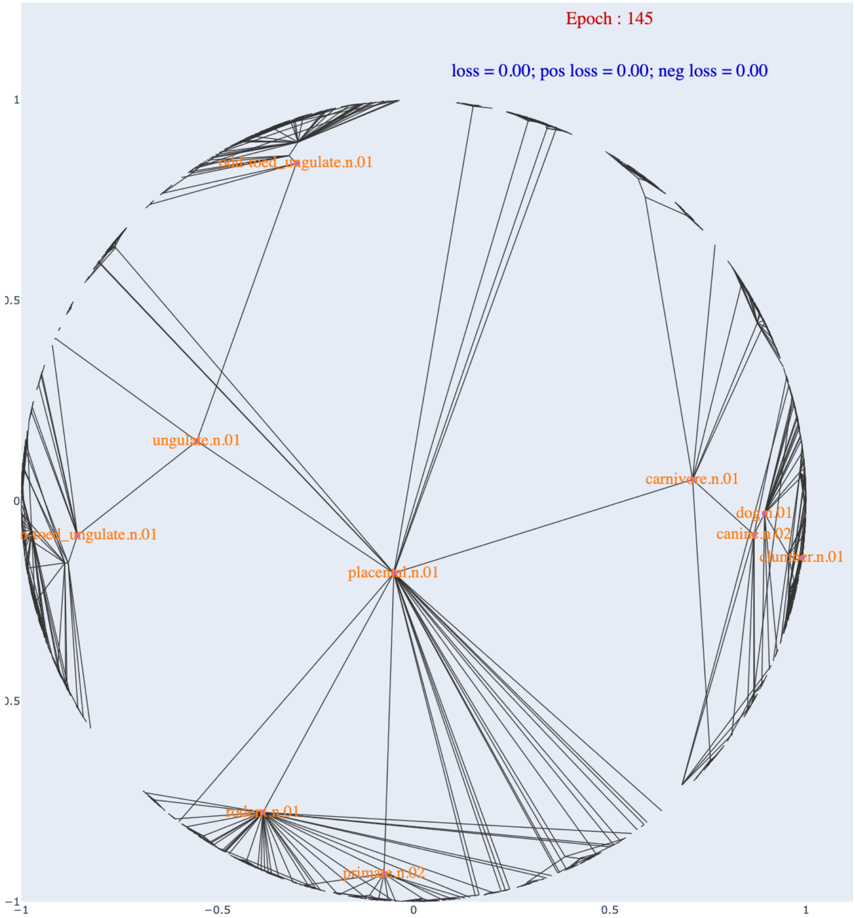
- Goal: given Poincaré embeddings of nouns, classify the nouns using MLR
- Three MLRs are compared: (1) hyperbolic, (2) Euclidean, and (3) Euclidean after applying the tangent mapping from the Poincaré manifold to the $\mathbb{R}^2$ plane

WordNet subtree	Model	d = 2	d = 3	d = 5	d = 10
animal.n.01 3218 / 798	Hyperbolic	$\mathbf{47.43 \pm 1.07\%}$	$\mathbf{91.92 \pm 0.61\%}$	$\mathbf{98.07 \pm 0.55\%}$	$\mathbf{99.26 \pm 0.59\%}$
	Direct Eucl	$41.69 \pm 0.19\%$	$68.43 \pm 3.90\%$	$95.59 \pm 1.18\%$	$\mathbf{99.36 \pm 0.18\%}$
	$\log_0$ + Eucl	$38.89 \pm 0.01\%$	$62.57 \pm 0.61\%$	$89.21 \pm 1.34\%$	$98.27 \pm 0.70\%$
group.n.01 6649 / 1727	Hyperbolic	$\mathbf{81.72 \pm 0.17\%}$	$\mathbf{89.87 \pm 2.73\%}$	$\mathbf{87.89 \pm 0.80\%}$	$\mathbf{91.91 \pm 3.07\%}$
	Direct Eucl	$61.13 \pm 0.42\%$	$63.56 \pm 1.22\%$	$67.82 \pm 0.81\%$	$\mathbf{91.38 \pm 1.19\%}$
	$\log_0$ + Eucl	$60.75 \pm 0.24\%$	$61.98 \pm 0.57\%$	$67.92 \pm 0.74\%$	$\mathbf{91.41 \pm 0.18\%}$
worker.n.01 861 / 254	Hyperbolic	$\mathbf{12.68 \pm 0.82\%}$	$\mathbf{24.09 \pm 1.49\%}$	$\mathbf{55.46 \pm 5.49\%}$	$\mathbf{66.83 \pm 11.38\%}$
	Direct Eucl	$10.86 \pm 0.01\%$	$22.39 \pm 0.04\%$	$35.23 \pm 3.16\%$	$47.29 \pm 3.93\%$
	$\log_0$ + Eucl	$9.04 \pm 0.06\%$	$22.57 \pm 0.20\%$	$26.47 \pm 0.78\%$	$36.66 \pm 2.74\%$
mammal.n.01 953 / 228	Hyperbolic	$\mathbf{32.01 \pm 17.14\%}$	$\mathbf{87.54 \pm 4.55\%}$	$\mathbf{88.73 \pm 3.22\%}$	$\mathbf{91.37 \pm 6.09\%}$
	Direct Eucl	$\mathbf{15.58 \pm 0.04\%}$	$44.68 \pm 1.87\%$	$59.35 \pm 1.31\%$	$77.76 \pm 5.08\%$
	$\log_0$ + Eucl	$13.10 \pm 0.13\%$	$44.89 \pm 1.18\%$	$52.51 \pm 0.85\%$	$56.11 \pm 2.21\%$

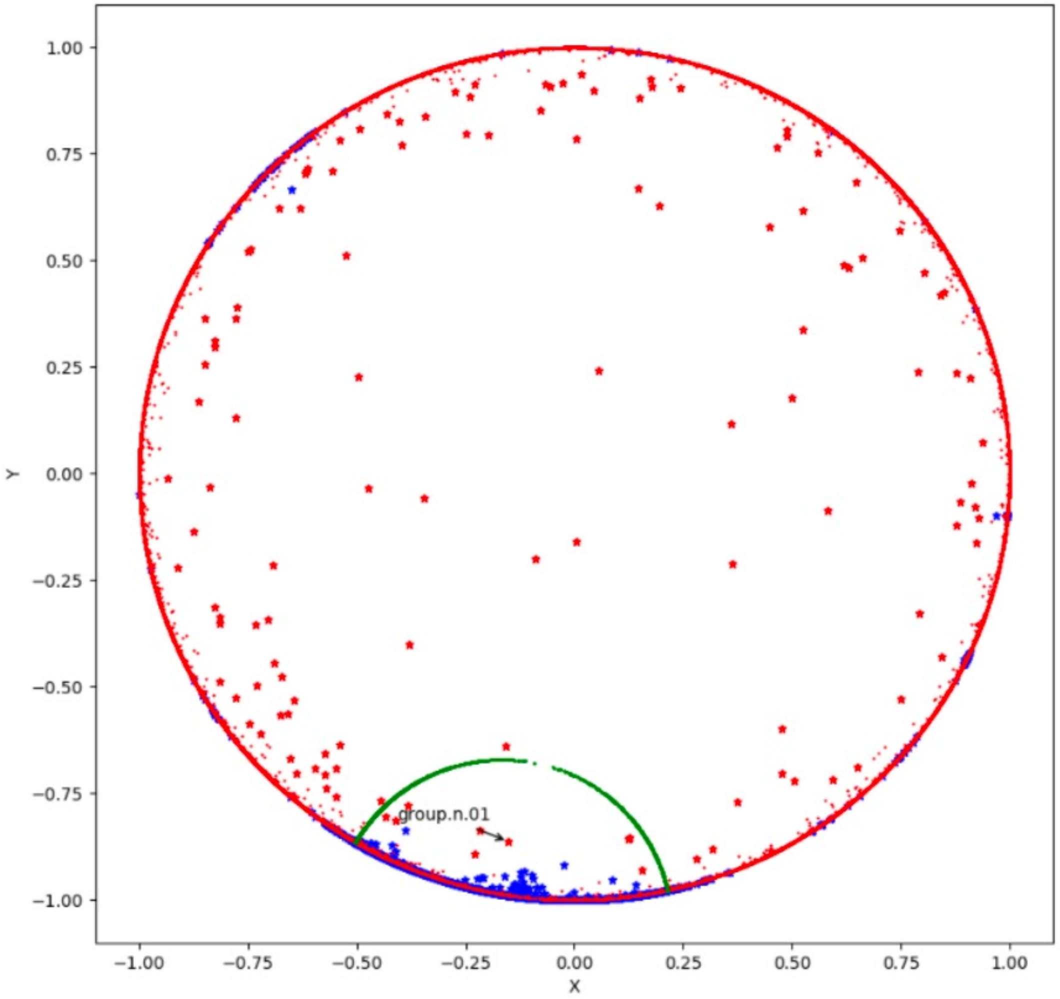
Poincaré Word Embeddings

Code output

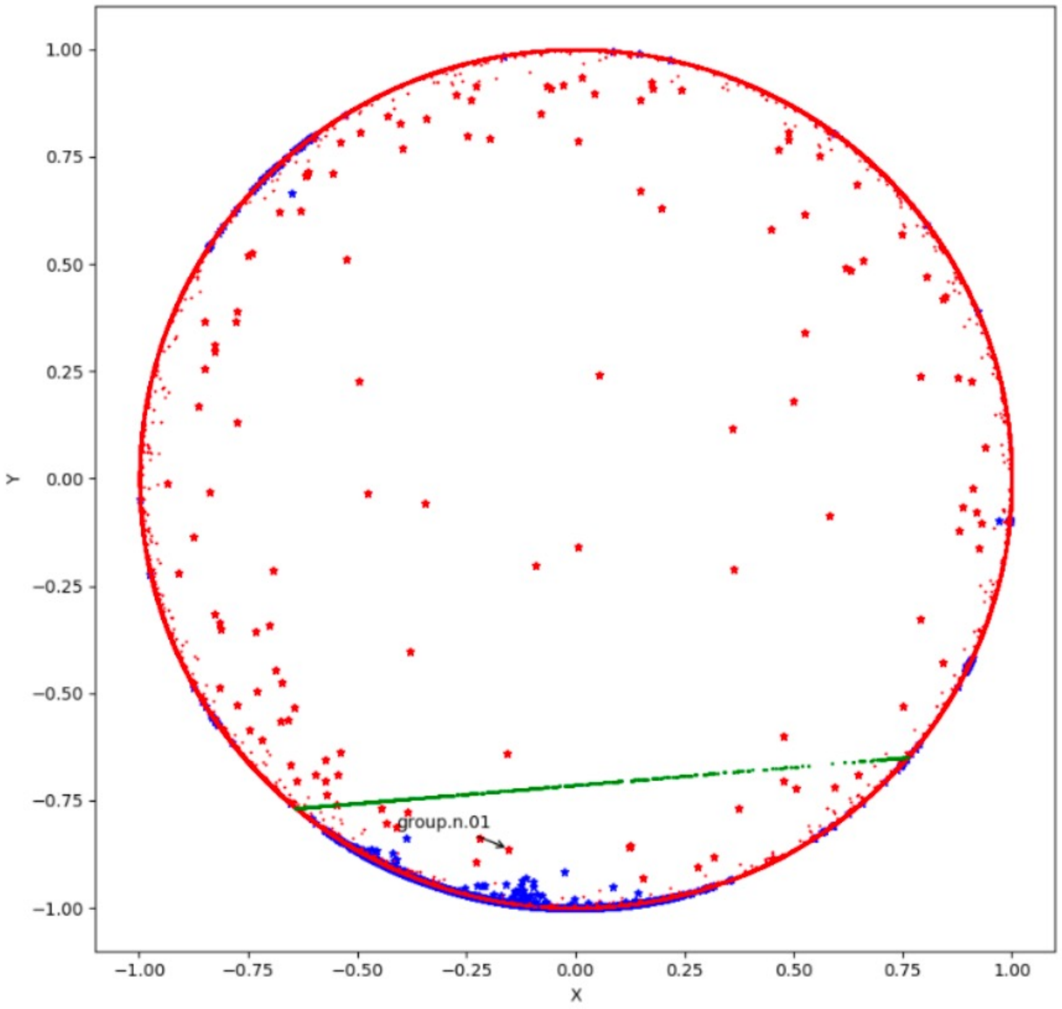




Experiment 2 MLR Results



Hyperbolic



Euclidean

Related Works

[1] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6341–6350, 2017.

[2] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *Proceedings of the thirty-fifth international conference on machine learning (ICML)*, 2018.

[3] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.

[4] John Lamping, Ramana Rao, and Peter Pirolli. A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408. ACM Press/Addison-Wesley Publishing Co., 1995.

[5] Abraham Albert Ungar. A gyrovector space approach to hyperbolic geometry. *Synthesis Lectures on Mathematics and Statistics*, 1(1):1–194, 2008.

[6] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6341–6350, 2017.