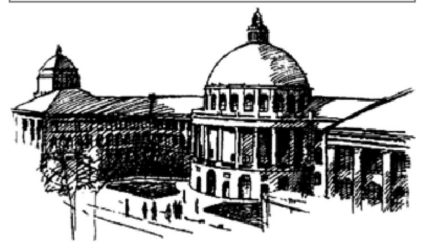
ETH zürich

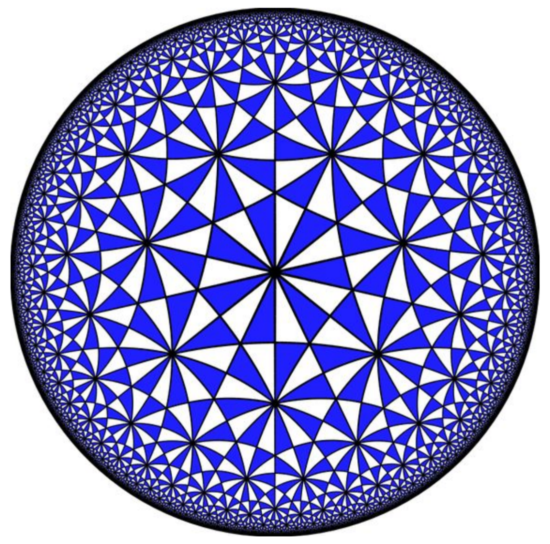


Hyperbolic Neural Networks

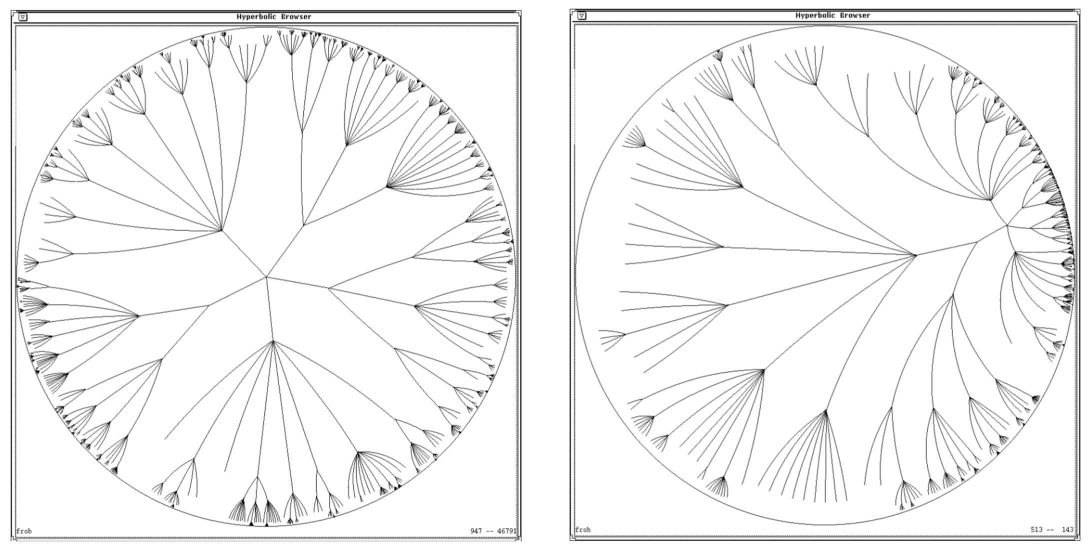
Octavian-Eugen Ganea*, Gary Bécigneul*, Thomas Hofmann

Department of Computer Science, ETH Zürich, Switzerland

Hyperbolic Geometry



- Hyperbolic space - constant negative curvature
- Non-euclidean embeddings
- Exponential volume growth (unlike polynomially in Euclidean space) ⇒ **Exponential capacity increase**
- Mathematically, can *isometrically* (preserve distances) embed:
  - approximate **tree-like** structures, or w/ **heterogeneous topology**
  - **scale-free** networks - node degree distributions follow a **power-law**



Taken from J. Lamping et al. "A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies." SIGCHI 1995.

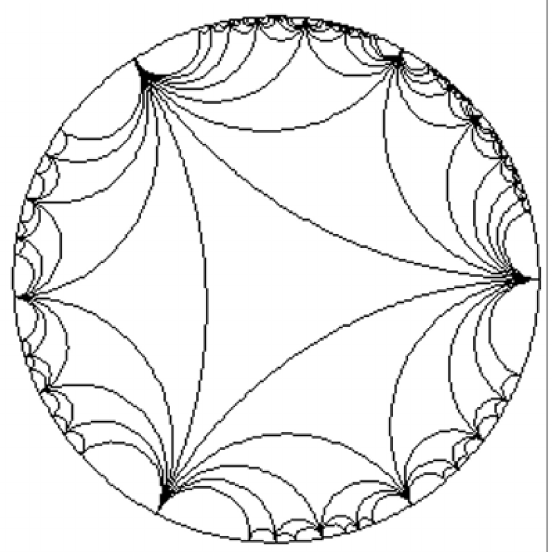
Hyperbolic Geometry in Machine Learning

Recently, hyperbolic embeddings in ML - e.g. Nickel & Kiela,2017

→ embed hierarchies: significantly superior **disentanglement behavior** than in Euclidean space due to the negative curvature

Difficulties

- HOW TO USE HYPERBOLIC EMBEDDINGS IN **DOWNSTREAM TASKS** ?
- HOW TO FEED HYPERBOLIC EMBEDDINGS TO **NEURAL NETS** ?



- basic Euclidean operations **not defined** in the hyperbolic space! *e.g. vector addition should follow hyperbolic "straight-lines", i.e. geodesics*
- neural networks **should not ignore** the hyperbolic geometry (e.g. hidden states of an RNN have to always be hyperbolic)

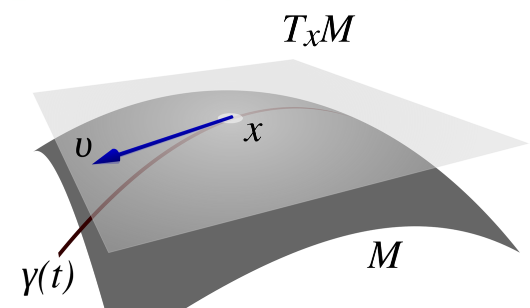
This work to the rescue :)

Our contributions

Use **Gyro-vector spaces** to *generalize basic operations* and *neural networks* from Euclidean to hyperbolic spaces:

- Gyro-vs: - analogue of Euclidean vector spaces
  - used in relativity theory (speeds of particles are hyperbolic)
- Vector addition $x + y \iff x \oplus_c y$
- Scalar multiplication $rx \iff r \otimes_c x$
  - Closed form distance $d_c(x, y) = (2/\sqrt{c}) \tanh^{-1}(\sqrt{c}\|-x \oplus_c y\|)$
  - Closed form geodesics: $\gamma_{x\to y}(t) := x \oplus_c (-x \oplus_c y) \otimes_c t$

1) We *connect Gyro-vs and Riemannian hyperbolic geometry*



- Closed form $\exp_x(v)$, $\log_x(y)$
- Closed form parallel transport (move across tangent spaces)

2) Hyperbolic Feed-forward Neural Networks

- *Möbius version of* $f : \mathbb{R}^n \to \mathbb{R}^m$ (e.g. pointwise non-linearity):

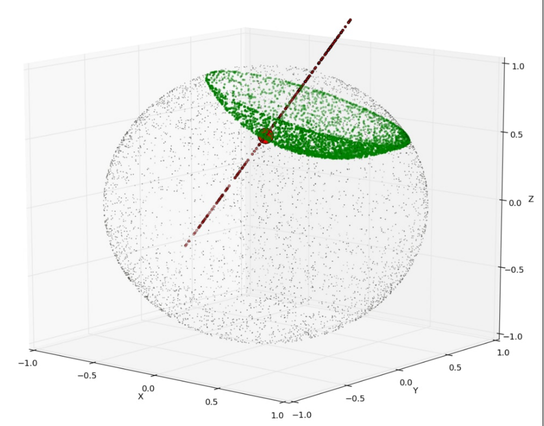
$$f^{\otimes_c} : \mathbb{D}^n_c \to \mathbb{D}^m_c, \quad f^{\otimes_c}(x) := \exp^c_{\mathbf{0}}(f(\log^c_{\mathbf{0}}(x)))$$

- Matrix - vector multiplication:

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

  - *Properties:* matrix associativity, scalar-matrix associativity, preserved rotations

3) Hyperbolic Softmax layer - Multiclass Logistic Regression



- Hyperbolic hyperplane:

$$\tilde{H}^c_{a,p} = \{x \in \mathbb{D}^n_c : \langle -p \oplus_c x, a\rangle = 0\}.$$

- **Theorem:** closed form of $d_c(x, \tilde{H}^c_{a,p})$
- Final MLR formula (based on Lebanon and Lafferty,2004):

$$p(y = k|x) \propto \exp\left(\frac{\lambda^c_{p_k}\|a_k\|}{\sqrt{c}} \sinh^{-1}\left(\frac{2\sqrt{c}\langle -p_k \oplus_c x, a_k\rangle}{(1 - c\|-p_k \oplus_c x\|^2)\|a_k\|}\right)\right)$$

Property: *All our models recover their Euclidean variants when curvature* $c \to 0$.

4) Hyperbolic Recurrent Networks, e.g. hGRU

$$\text{hyp-GRU} \leftarrow \begin{cases} r_t = \sigma \log^c_{\mathbf{0}}(W^r \otimes_c h_{t-1} \oplus_c U^r \otimes_c x_t \oplus_c b^r) \\ \tilde{h}_t = \varphi^{\otimes_c}((W \mathrm{diag}(r_t)) \otimes_c h_{t-1} \oplus_c U \otimes_c x_t \oplus b) \\ h_t = h_{t-1} \oplus_c \mathrm{diag}(z_t) \otimes_c (-h_{t-1} \oplus_c \tilde{h}_t) \end{cases}$$

- Hyperbolic hidden states
- **Theorem:** update-gate mechanism derived from *time-warping invariance* principle (via gyro-derivative and gyro-chain-rule)

Experiments

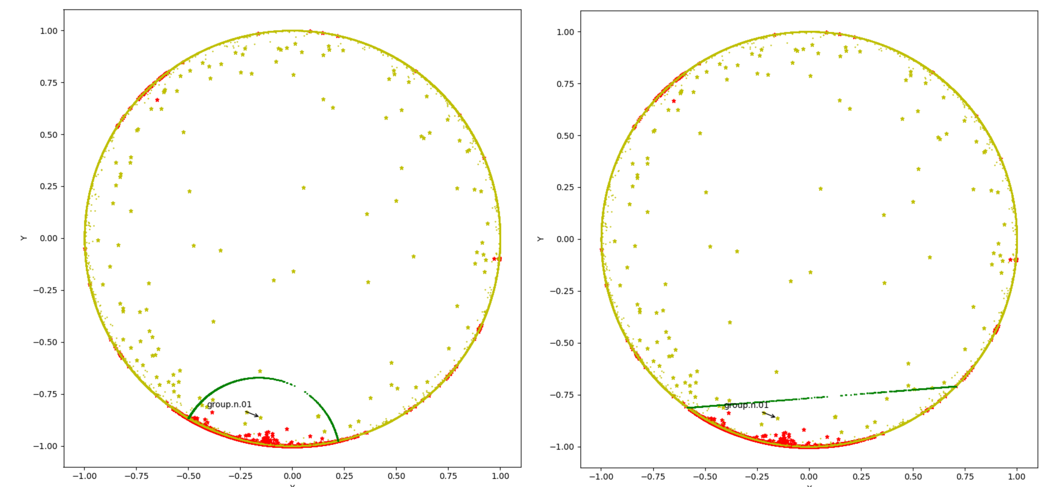
1) Textual Entailment tasks (semantic + syntactic).

TEST ACCURACY	SNLI	PREFIX-10%	PREFIX-30%	PREFIX-50%
FULLY EUCLIDEAN RNN	**79.34** %	89.62 %	81.71 %	72.10 %
HYP 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 %
HYP GRU+FFNN, EUCL MLR	79.76 %	**97.36** %	**88.47** %	**76.87** %
FULLY HYPERBOLIC GRU	**81.19** %	**97.14** %	**88.26** %	**76.44** %

2) MLR experiments.

Test F1 classification scores (%) for 4 subtrees of WordNet tree.

WORDNET SUBTREE	MODEL	D = 2	D = 3	D = 5	D = 10
ANIMAL.N.01 3218 / 798	HYP	$\mathbf{47.43 \pm 1.07}$	$\mathbf{91.92 \pm 0.61}$	$\mathbf{98.07 \pm 0.55}$	$\mathbf{99.26 \pm 0.59}$
	EUCL	41.69 ± 0.19	68.43 ± 3.90	95.59 ± 1.18	$\mathbf{99.36 \pm 0.18}$
	$\log_{\mathbf{0}}$	38.89 ± 0.01	62.57 ± 0.61	89.21 ± 1.34	98.27 ± 0.70
GROUP.N.01 6649 / 1727	HYP	$\mathbf{81.72 \pm 0.17}$	$\mathbf{89.87 \pm 2.73}$	$\mathbf{87.89 \pm 0.80}$	$\mathbf{91.91 \pm 3.07}$
	EUCL	61.13 ± 0.42	63.56 ± 1.22	67.82 ± 0.81	$\mathbf{91.38 \pm 1.19}$
	$\log_{\mathbf{0}}$	60.75 ± 0.24	61.98 ± 0.57	67.92 ± 0.74	$\mathbf{91.41 \pm 0.18}$
WORKER.N.01 861 / 254	HYP	$\mathbf{12.68 \pm 0.82}$	$\mathbf{24.09 \pm 1.49}$	$\mathbf{55.46 \pm 5.49}$	$\mathbf{66.83 \pm 11.38}$
	EUCL	10.86 ± 0.01	22.39 ± 0.04	35.23 ± 3.16	47.29 ± 3.93
	$\log_{\mathbf{0}}$	9.04 ± 0.06	22.57 ± 0.20	26.47 ± 0.78	36.66 ± 2.74
MAMMAL.N.01 953 / 228	HYP	$\mathbf{32.01 \pm 17.14}$	$\mathbf{87.54 \pm 4.55}$	$\mathbf{88.73 \pm 3.22}$	$\mathbf{91.37 \pm 6.09}$
	EUCL	$\mathbf{15.58 \pm 0.04}$	44.68 ± 1.87	59.35 ± 1.31	77.76 ± 5.08
	$\log_{\mathbf{0}}$	13.10 ± 0.13	44.89 ± 1.18	52.51 ± 0.85	56.11 ± 2.21



Hyperbolic (left) vs Direct Euclidean (right) binary MLR used to classify nodes as being part in the GROUP.N.01 *subtree of the WordNet noun hierarchy solely based on their Poincaré embeddings.*