HYPERBOLIC WORD EMBEDDINGS

Authors: Alexandru Tifrea ,
Gary Becigneul,
Octavian-Eugen Ganea

Abstract

- Words have varying importance and hidden connections.

- Researchers have a new method to understand words without human guidance.

- They use a special math-based way to represent words.

- This method outperforms existing techniques in various word-related tasks, including classification.

Introduction

1. Word embeddings are essential for natural language processing, converting words into a format suitable for deep learning models.

2. Popular models like Glove, Word2Vec, and FastText learn word vectors from raw text data based on co-occurrence statistics.

3. The proposed approach combines point embeddings and Gaussian embeddings, addressing the limitations of both.

4. The new method performs well in word similarity, analogy, and hypernymy tasks parallely.

Recent Approaches in Embedding Graphs and Hierarchies

- Recent methods aim to embed graphs into low-dimensional spaces to improve link prediction, using order, hyperbolic geometry, or both.

- Learning word embeddings with hierarchical info is done through supervised and unsupervised methods, which may rely on external data like WordNet.

- Some recent attempts to learn unsupervised word embeddings in hyperbolic space face issues, such as poor performance, asymmetric relation modeling, and small training datasets.

- The authors intend to address these challenges and improve hypernymy detection in unsupervised word embeddings by connecting with density-based methods.

Hyperbolic Spaces And Their Cartesian Product

- When working with hyperbolic space, there are several mathematical models available. In this case, we have selected one particular model out of five isometric models. The chosen model is the Poincaré ball for representing Hyperbolic space, denoted as $\mathbb{D}^n = \{x \in \mathbb{R}^n \mid \|x\|_2 < 1\}$ In simpler terms, this represents a ball in n-dimensional space where the distance of any point inside the ball from the origin represented by (x) is less than 1.

Isometric deformation φ of $\mathbb{D}^2$ into $\mathbb{H}^2$.

Hyperbolic Spaces And Their Cartesian Product

- This illustrates the concept in two dimensions (n=2). the dark lines in the Poincaré ball represent geodesics, which are the shortest paths between two points in a curved space like a sphere or, in this case, a hyperbolic space.

- This introduces the distance function for measuring distances between points (x) and (y) in the Poincaré ball. The distance function in this hyperbolic space is given by the equation (cosh^{-1}), which is the inverse hyperbolic cosine function. This function calculates the distance between two points x and y in the Poincaré ball.

Euclidean GLoVe

- GloVe (Global Vectors for Word Representation) is an unsupervised learning algorithm for obtaining vector representations for words.
- Learning word representations in the Euclidean space from statistics of word co-occurrences in a text corpus,
- Geometrically capture the words meaning and relations.

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left(-h(d(w_i, \tilde{w}_j)) + b_i + \tilde{b}_j - \log X_{ij} \right)^2$$

V = size of the vocabulary

f = down-weights the signal coming from frequent words

W_i = target word

$\tilde{w}_j$ = *context word*

b =bais

h = function to be chosen as a hyperparameter of the model

d = can be any differentiable distance function

Gaussian Embeddings

- Represent words in Gaussian.
- Use Mean Vector (μ) and Covariance Matrix (σ)
- Covariance is tells how spread specific words meaning are.

Hyperbolic Embeddings

- Hyperbolic spaces better for representing words in tree like structure
- Great for capturing hierarchy and relationships

Analogies For Hyperbolics

- Finding relationships between words in a word embedding space. *For example : "Paris" is related to "France" as "Tokyo" is related to "Japan"*
- d = c + (b - a) = b + (c - a) – a, b, c, and d are word embeddings.

Challenges with Gaussian Embeddings

- Word as Gaussian distributions.
- Not straightforward to perform analogy computations

Solution: Hyperbolic Geometry

- connecting Gaussian embeddings and hyperbolic embeddings.
- In hyperbolic space, "analogy parallelograms" are naturally defined, as

  $d1 = c \oplus gyr[c, a](a \oplus b)$ and **$d2 = b \oplus gyr[b, a](a \oplus c)$**

- d1 and d2 differ is due to the curvature of the hyperbolic space.

For evaluation,

$$m^{t}_{d1d2} := d1 \oplus ((-d1 \oplus d2) \otimes t) \quad \text{for } t \in [0, 1]$$

- if t = 1/2, this is called the gyro-midpoint and then $m^{0.5}_{d1d2} = m^{0.5}_{d2d1}$ which is at equal hyperbolic distance from d1 as from d2.
- Measure analogy relationships at different points along the path.
- Continuously deforming the hyperbolic space into the Euclidean space allows to recover analogy computations as they would appear in a standard Euclidean space.

Fisher geometry

- Describes Gaussian Embeddings and is hyperbolic

Fisher distance between two distributions relates to the hyperbolic distance in $\mathbb{H}^2$:

$$d_F\left(\mathcal{N}(\mu,\sigma^2),\mathcal{N}(\mu',\sigma'^2)\right) = \sqrt{2}d_{\mathbb{H}^2}\left((\mu/\sqrt{2},\sigma),(\mu'/\sqrt{2},\sigma')\right).$$

For n-dimensional Gaussians with diagonal covariance matrices written $\Sigma = \mathrm{diag}(\sigma)^2$, it becomes:

$$d_F\left(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu',\Sigma')\right) = \sqrt{\sum_{i=1}^{n} 2d_{\mathbb{H}^2}\left((\mu_i/\sqrt{2},\sigma_i),(\mu_i'/\sqrt{2},\sigma_i')\right)^2}.$$

Hence there is a direct correspondence between diagonal Gaussians and the product space $(\mathbb{H}^2)^n$.

Fisher Distance and KL (Kullback-Leibler) Divergence

- measuring the two probability distributions.
- compare how similar or dissimilar two Gaussian distributions.

Riemannian optimization

- RADAGRAD : Adjusting and fine-tuning the embeddings in hyperbolic spaces
- Provides better results

Towards A Principled Score For Entailment / Hypernymy

1. Defining Generic and Specific Sets:

   The procedure begins by defining two sets of words: G (generic) and S (specific). These sets represent word embeddings in the hyperbolic space that are believed to be generic and specific, respectively.

2. Isometry for Alignment:

   The correct isometry (geometric transformation) is applied to the word embeddings to align them correctly for comparison. This alignment is essential to ensure that the is-a score is calculated accurately.

3. Translation and Rotation : The embeddings undergo two operations:

- Mobius Translation: All word embeddings are translated by a global mean, which is computed as the average of the means of the generic and specific word embeddings. This translation aligns the embeddings correctly.
- Rotation: The embeddings are rotated so that a specific word is mapped to a specific point in the hyperbolic space (in this case, (0, 1)).

4. Coordinate Conversion:

   The embeddings are converted from Poincaré disk coordinates to half-plane coordinates. The authors refer to this conversion as "poincare2halfplane."

5. Mapping to Gaussian Parameters:

   The half-plane coordinates are further transformed into Gaussian parameters, specifically, mean (μ) and variance (σ). This mapping is based on specific mathematical formulas. Each coordinate in the half-plane is used to calculate the corresponding mean and variance for the Gaussian distribution.

6. Is-a Score Calculation:

   The final step involves calculating the is-a score between the two sets of embeddings (generic and specific) using the Gaussian parameters. The is-a score is calculated for each coordinate and is the difference between the logarithms of the variances of the corresponding Gaussian distributions.

Embedding Symbolic Data In A Continuous Space With Matching Hyperbolicity

- δ-Hyperbolicity:

  This number helps us understand how much a space is like a tree. A low number means the space is tree-like, which can be good for understanding word connections.

- Computing δavg:

  To find this δ-hyperbolicity number for their data, we need to calculate the distances between pairs of words. But they only have information about how similar words are, not how far apart they are.

$h(x)$	$\log(x)$	x	x^2	$\cosh(x)$	$\cosh^2(x)$	$\cosh^4(x)$	$\cosh^5(x)$	$\cosh^{10}(x)$
d_{avg}	18950.4	18.9505	4.3465	3.68	2.3596	1.7918	1.6888	1.4947
δ_{avg}	8498.6	0.7685	0.088	0.0384	0.0167	0.0072	0.0056	0.0026
$2\delta_{avg}/d_{avg}$	0.8969	0.0811	0.0405	0.0209	0.0142	0.0081	0.0066	0.0034

Table 1: average distances, δ-hyperbolicities and ratios computed via sampling for the metrics induced by different h functions, as defined in Eq. (7).

The results suggest that the discrete metric spaces obtained from their symbolic data of co-occurrences have low hyperbolicity. This implies that embedding words in hyperbolic spaces, or products of hyperbolic spaces, could be a suitable approach.

Experiments: Similarity, Analogy,Entailment

1.Dataset: Trained word embedding models on an English Wikipedia corpus with 1.4 billion tokens, filtering out words appearing less than 100 times.

2.Model Types: Used Poincare models and Euclidean GloVe models for similarity and analogy tasks.

3.Hyperbolic Models: Employed two different hyperbolic functions for Poincare models: $h(x) = x^2$ and $h(x) = \cosh^2(x)$.

4.Learning Rates: Euclidean GloVe and Poincare models with $h(x) = x^2$ used learning rate 0.05, while Poincare models with $h(x) = \cosh^2(x)$ used learning rate 0.01.

5.Similarity Results: Poincare models generally outperformed vanilla GloVe models in word similarity tasks.

Table 2: Word similarity results for 100-dimensional models. Highlighted: the **best** and the 2^{nd} best.

Experiment name	RareWord	WordSim	SimLex	SimVerb	MC	RG
100D Vanilla GloVe	0.3798	0.5901	0.2963	0.1675	0.6524	0.6894
100D Vanilla GloVe w/ init trick	0.3787	0.5668	0.2964	0.1639	0.6562	0.6757
100D Poincaré GloVe $h(x) = \cosh^2(x)$, w/ init trick	0.4187	0.6209	**0.3208**	**0.1915**	0.7833	0.7578
50x2D Poincaré GloVe $h(x) = \cosh^2(x)$, w/ init trick	**0.4276**	**0.6234**	0.3181	0.189	**0.8046**	**0.7597**
50x2D Poincaré GloVe $h(x) = x^2$, w/ init trick	0.4104	0.5782	0.3022	0.1685	0.7655	0.728

This table provides a comparative evaluation of word embedding models, both in Euclidean and hyperbolic spaces.

Table 3: Nearest neighbors (in terms of Poincaré distance) for some words using our 100D hyperbolic embedding model.

sixties	seventies, eighties, nineties, 60s, 70s, 1960s, 80s, 90s, 1980s, 1970s
dance	dancing, dances, music, singing, musical, performing, hip-hop, pop, folk, dancers
daughter	wife, married, mother, cousin, son, niece, granddaughter, husband, sister, eldest
vapor	vapour, refrigerant, liquid, condenses, supercooled, fluid, gaseous, gases, droplet
ronaldo	cristiano, ronaldinho, rivaldo, messi, zidane, romario, pele, zinedine, xavi, robinho
mechanic	electrician, fireman, machinist, welder, technician, builder, janitor, trainer, brakeman
algebra	algebras, homological, heyting, geometry, subalgebra, quaternion, calculus, mathematics, unital, algebraic

This table showcases the model's ability to capture semantic relationships and similarities between words in a hyperbolic embedding space.

Table 4: Word analogy results for 100-dimensional models. Highlighted: the **best** and the 2^{nd} best.

Experiment name	Method	SemGoogle	SynGoogle	Google	MSR
100D Vanilla GloVe	3COSADD	0.6005	0.5869	0.5931	0.4868
100D Vanilla GloVe w/ init trick	3COSADD	0.6427	0.595	0.6167	0.4826
100D Poincaré GloVe $h(x) = \cosh^2(x)$, w/ init. trick	Cosine dist	**0.6641**	**0.6088**	**0.6339**	**0.4971**
50x2D Poincaré GloVe $h(x) = x^2$, w/ init. trick	Poincaré dist	0.6582	0.6066	0.6300	0.4672
50x2D Poincaré GloVe $h(x) = \cosh^2(x)$, w/ init. trick	Poincaré dist	0.6048	0.6042	0.6045	0.4849

The table demonstrates the performance of different word embedding models in solving word analogies.

Hyperbolic word embeddings, specifically Poincare GloVe, consistently outperform the traditional GloVe model across various analogy datasets.

Hypernymy results discussion

1. In this section demonstrating that their hyperbolic word embeddings are capable of capturing hierarchical relationships between words effectively, even in an unsupervised setting.
2. Hyperbolic embeddings capture hierarchical word relationships effectively without the need for extensive supervision.

Conclusion

1. The paper concludes by proposing the adaptation of GloVe for hyperbolic spaces, leveraging the connection to Gaussian distributions.
2. It justifies the hyperbolic spaces
3. This is the first model that excels in word similarity, analogy, and hypernymy detection simultaneously

— THANK YOU —