Workshop on SNL (CC-BY-SA 4.0)

Distributed Representations of Compositional Structures

Methods using Circular Convolution

Katsuhiko Hayashi

Hokkaido University
Faculty of Information Science and Technology
katsuhiko-h@ist.hokudai.ac.jp

2023-06-29

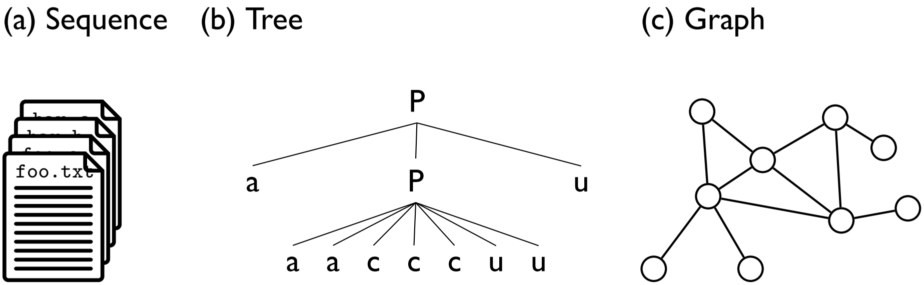
TODAY'S AGENDA

- Background
- Representation Learning with HRR
- Circular HRR and Binary Spatter Codes

TODAY'S AGENDA

- Background
- Representation Learning with HRR
- Circular HRR and Binary Spatter Codes

Background: Structured Data

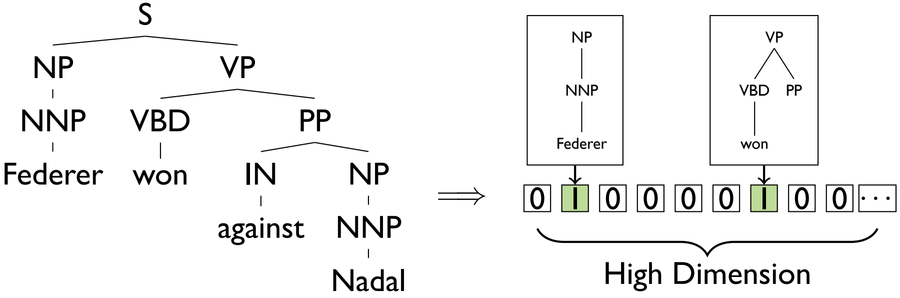


(a) Sequence (b) Tree (c) Graph

Distributed Representations are attractive for modeling structured data:

- **Similarity**: Convolution Kernel, Representation Learning, Semantic Hashing
- **Classification**: Multi-label Classification, Structured Prediction

(I) Feature Extraction from Trees

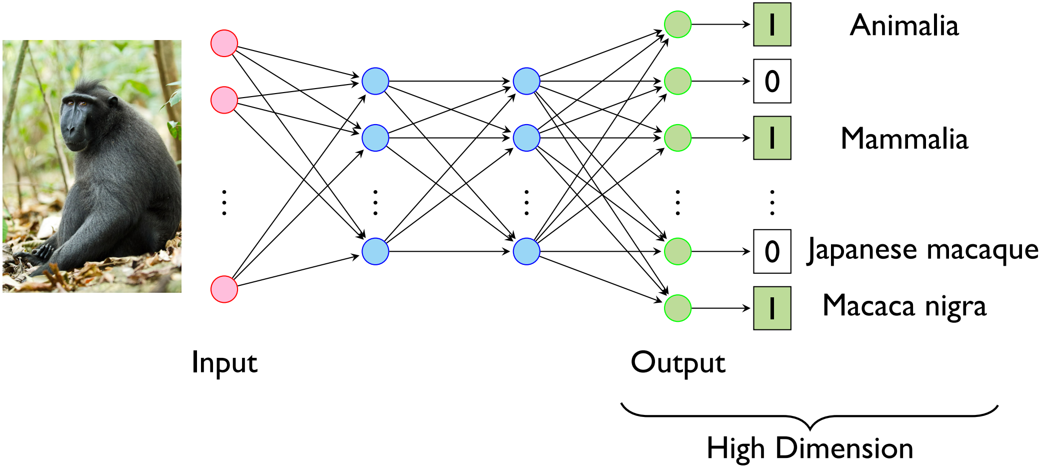


Trees are fundamental data structures used to represent very different objects such as proteins, HTML documents, or NL utterances

Tree Kernel [NIPS2001] requires extracting all tree fragments that occur in a tree

- Dynamic Programming, Distributed Tree Kernel [ICML2011]

(2) Multi-label Classification



1 Animalia

0

1 Mammalia

⋮

0 Japanese macaque

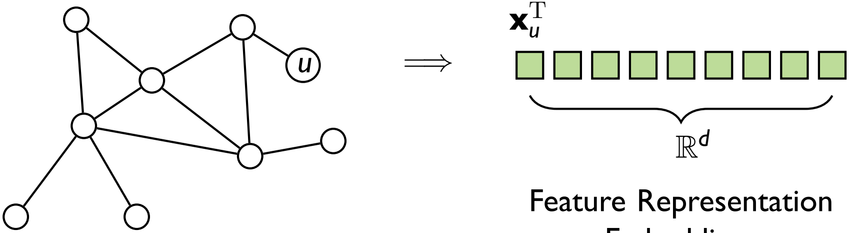
1 Macaca nigra

Input

Output

High Dimension

(3) Representation Learning on Graphs

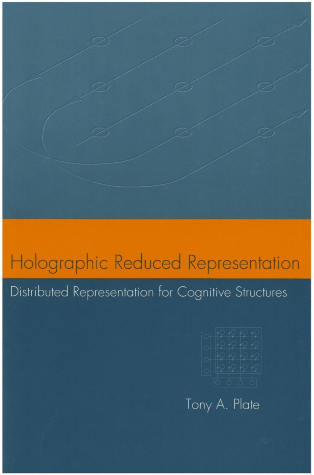


Feature Representation
Embeddings

Graph Representation Learning is to construct a set of features ("embeddings") representing the structure (nodes/edges) of the graph

- node classification, link prediction, etc.

Holographic Reduced Representations


Holographic Reduced Representation
Distributed Representation for Cognitive Structures
Tony A. Plate

HRR is a method for representing compositional structure in distributed representation [IJCAI1991]

Eg.: Frame Structure “Mark eats the fish”

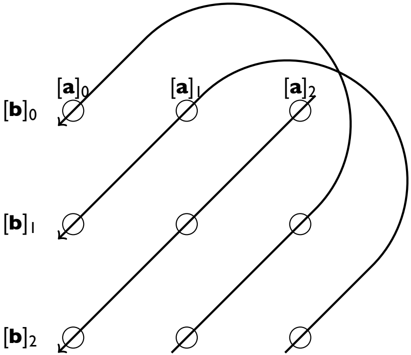
$$\mathbf{f} = \mathbf{eat} + \mathbf{agt}_{eat} * \mathbf{Mark} + \mathbf{obj}_{eat} * \mathbf{Fish}$$

where $*$ is an encoding operator

HRR has a decoding operator

$$\mathbf{agt}_{eat} \star \mathbf{f} \approx \mathbf{Mark}$$

HRR Encoding Operator $*$



$$[\mathbf{a} * \mathbf{b}]_j = \sum_{k=0}^{d-1} [\mathbf{a}]_k [\mathbf{b}]_{(j-k)}$$

$$(\text{for } j = 0, \ldots, d-1)$$

Circulant Matrix of a vector:

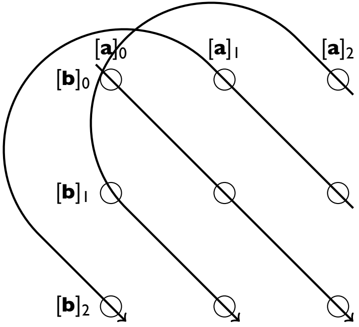
$$circ(\mathbf{a}) = \begin{bmatrix} [\mathbf{a}]_0 & [\mathbf{a}]_2 & [\mathbf{a}]_1 \\ [\mathbf{a}]_1 & [\mathbf{a}]_0 & [\mathbf{a}]_2 \\ [\mathbf{a}]_2 & [\mathbf{a}]_1 & [\mathbf{a}]_0 \end{bmatrix}$$

Circulant Matrix $circ(\mathbf{a}) \in \mathbb{R}^{d \times d}$ can be written as $\mathbf{F}_d^{-1} \operatorname{diag}(\mathbf{F}_d \mathbf{a}) \mathbf{F}_d$ where $\mathbf{F}_d$ is the $d \times d$ Fourier matrix

Circular Convolution $*$:

$$\mathbf{a} * \mathbf{b} = circ(\mathbf{a})\mathbf{b} = \mathbf{F}_d^{-1}(\mathbf{F}_d \mathbf{a} \odot \mathbf{F}_d \mathbf{b})$$

HRR Decoding Operator $\star$



$$[\mathbf{a} \star \mathbf{b}]_j = \sum_{k=0}^{d-1} [\mathbf{a}]_k [\mathbf{b}]_{(k+j)}$$

$$(\text{for } j = 0, \ldots, d-1)$$

The transpose of a circulant matrix of a vector:

$$circ(\mathbf{a})^{\mathrm{T}} = \begin{bmatrix} [\mathbf{a}]_0 & [\mathbf{a}]_1 & [\mathbf{a}]_2 \\ [\mathbf{a}]_2 & [\mathbf{a}]_0 & [\mathbf{a}]_1 \\ [\mathbf{a}]_1 & [\mathbf{a}]_2 & [\mathbf{a}]_0 \end{bmatrix}$$

Circular Correlation $\star$:

$$\mathbf{a} \star \mathbf{b} = circ(\mathbf{a})^{\mathrm{T}} \mathbf{b} = \mathbf{F}_d^{-1} (\overline{\mathbf{F}_d \mathbf{a}} \odot \mathbf{F}_d \mathbf{b})$$

$\overline{\mathbf{z}}$ represents the complex conjugate of a complex vector $\mathbf{z}$

Why Correlation decodes Convolution?

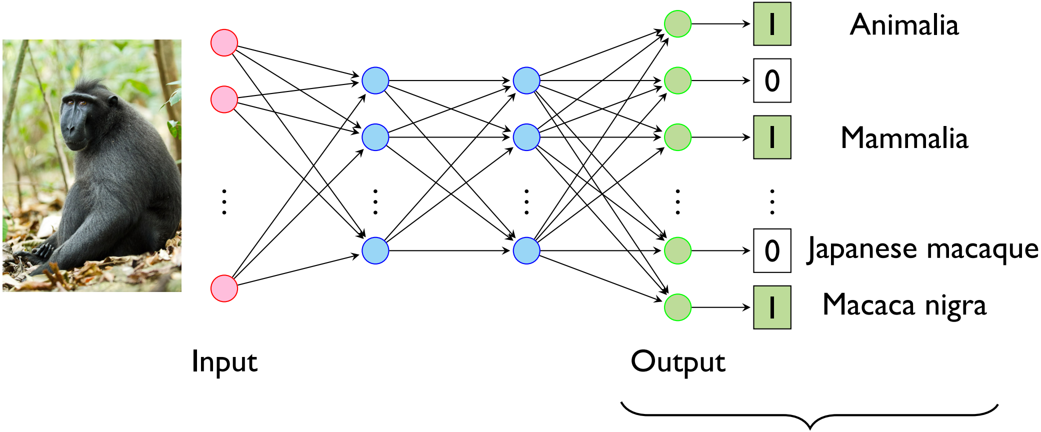
Consider 3-dim. vectors $\mathbf{c} = [c_0, c_1, c_2]^{\mathrm{T}}$ and $\mathbf{x} = [x_0, x_1, x_2]^{\mathrm{T}}$ where c_i and x_i are independently drawn from $N(0, \frac{1}{d})$ (here, $d = 3$). The convolution of $\mathbf{x}$ and $\mathbf{c}$ is

$$\mathbf{m} = \mathbf{c} * \mathbf{x} = \begin{bmatrix} c_0x_0 + c_1x_2 + c_2x_1 \\ c_0x_1 + c_1x_0 + c_2x_2 \\ c_0x_2 + c_1x_1 + c_2x_0 \end{bmatrix}$$

We could reconstruct $\mathbf{x}$ from this trace $\mathbf{m}$ by correlating with the cue $\mathbf{c}$

$$\mathbf{c} \star \mathbf{c} * \mathbf{x} = \begin{bmatrix} x_0(1+\zeta) + \eta_0 \\ x_1(1+\zeta) + \eta_1 \\ x_2(1+\zeta) + \eta_2 \end{bmatrix} \approx \mathbf{x} \quad \text{where } \zeta \overset{\mathrm{d}}{=} N(0, \tfrac{2}{d}) \text{ and } \eta_i \overset{\mathrm{d}}{=} N(0, \tfrac{d-1}{d^2})$$

Multi-label Learning with HRR (1/4)



1 Animalia

0

1 Mammalia

0 Japanese macaque

1 Macaca nigra

Input

Output

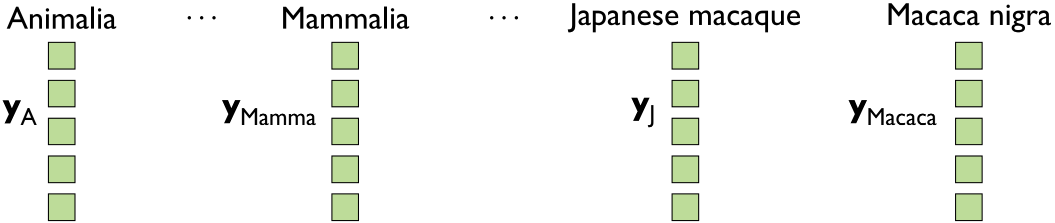
High Dimension

Multi-label Learning with HRR (2/4)

“Learning with Holographic Reduced Representations”,
Ganesan et. al., NeurIPS, 2021.

We begin by converting each class label into a label vector $\mathbf{y} \in \mathbb{R}^d$

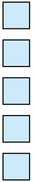
- ► d is much smaller than the size of the set of output classes: $d \ll L$
- ► All vector elements are sampled according to $N(0, \frac{1}{d})$



Multi-label Learning with HRR (3/4)

“Learning with Holographic Reduced Representations”,
Ganesan et. al., NeurIPS, 2021.

Using the HRR framework, we construct a **label vector $\mathbf{y}_x$** for a given data *x*

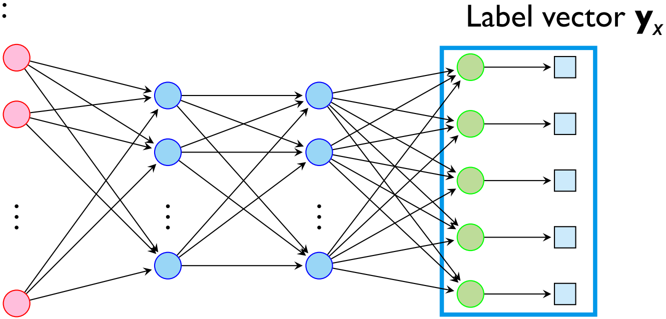


$$\mathbf{y}_x = \mathbf{y}_P * \underbrace{(\mathbf{y}_{\mathrm{A}} + \mathbf{y}_{\mathrm{Mamma}} + \cdots + \mathbf{y}_{\mathrm{Macaca}})}_{\mathbf{t}_P} + \mathbf{y}_N * \underbrace{(\cdots + \mathbf{y}_{\mathrm{J}})}_{\mathbf{t}_N}$$

Multi-label Learning with HRR (4/4)

"Learning with Holographic Reduced Representations",
Ganesan et. al., NeurIPS, 2021.

Training Phase:

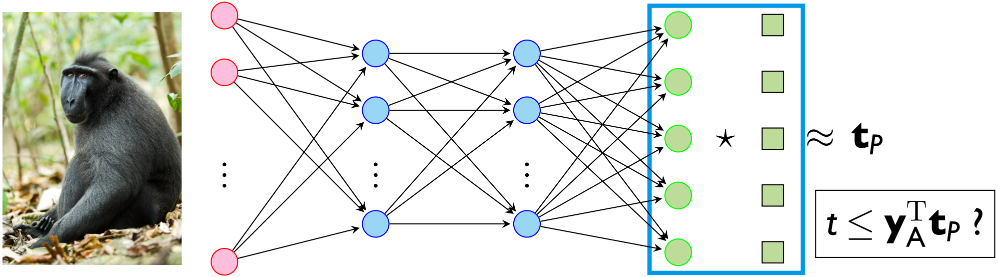


Multi-label Learning with HRR (4/4)

“Learning with Holographic Reduced Representations”,
Ganesan et. al., NeurIPS, 2021.

Inference Phase:

Key vector $\mathbf{y}_P$



$\star$

$\approx \mathbf{t}_P$

$$\boxed{t \le \mathbf{y}_A^{\mathrm{T}} \mathbf{t}_P \text{ ?}}$$

Why Addition Memories work well?

One way to store k vectors is to add them together: $\mathbf{t} = \mathbf{a} + \mathbf{b} + \mathbf{c}$ ($k = 3$)

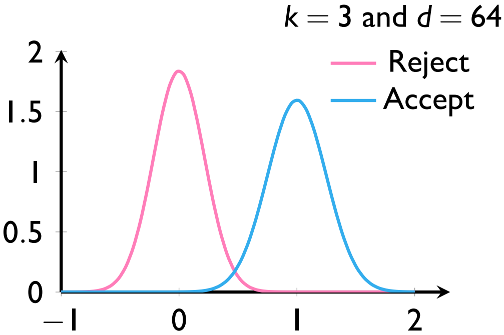
Such storage does allow for determining whether an item has been stored or not

To test whether a vector $\mathbf{x}$ is in the trace $\mathbf{t}$, we compute the dot product:

$$\mathbf{x} = \mathbf{a} \quad s_{Accept} = \mathbf{a}^{\mathrm{T}}\mathbf{t}$$
$$\mathbf{x} = \mathbf{d} \quad s_{Reject} = \mathbf{d}^{\mathrm{T}}\mathbf{t}$$

s_{Accept} and s_{Reject} are distributed as:

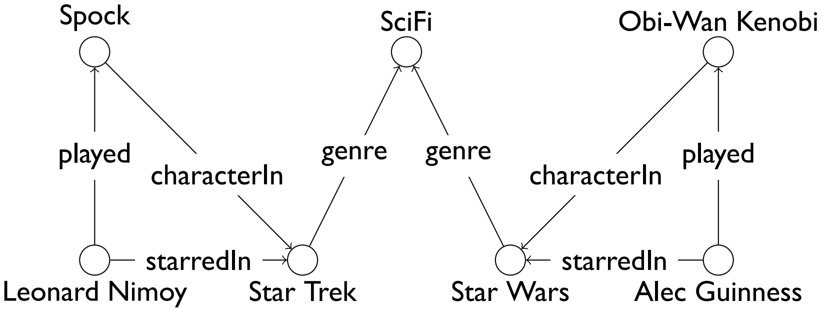
$$s_{Accept} \overset{d}{=} N(1, \frac{(k+1)}{d})$$
$$s_{Reject} \overset{d}{=} N(0, \frac{k}{d})$$



TODAY'S AGENDA

- ▶ Background
- ▶ Representation Learning with HRR
- ▶ Circular HRR and Binary Spatter Codes

Definition: RDF Knowledge Base

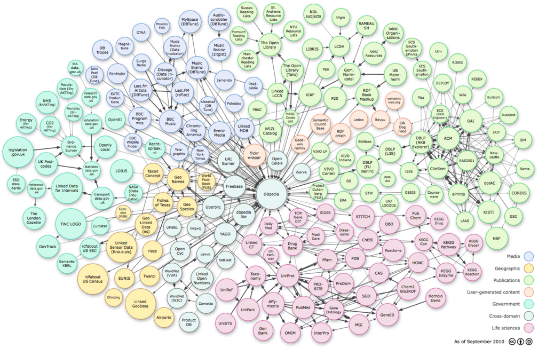


KB $\mathcal{K}$ is a set of facts

- **fact**: a triple of the form (s, r, o) (alternative notation $r(s, o)$)
- s/o is the **subject/object**, r is the **relation** (or predicate)

KB Example: DBPedia

DBPedia [Semantic Web2013]: A project aiming to extract structured content from Wikipedia information



https://www.dbpedia.org/

Open Knowledge Bases

The research on Semantic Web and Linked Data led to many open datasets

These open datasets are rebranded as “Knowledge Graphs”

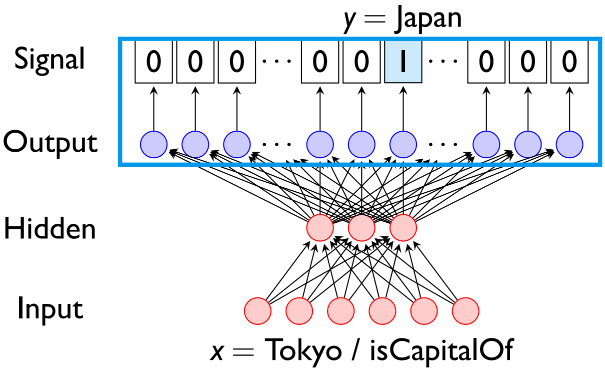
- DBPedia, Freebase, YAGO, NELL, Wikidata, KBPedia, Datacommons.org

Many open Knowledge Bases are sourced from Wikipedia and also benefited from unstructured corpus for their building process

Problem: KBs are imcomplete and have many missing links

- 93.8% of persons from Freebase have no place of birth and 78.5% have no nationality

Representation Learning on KBs



Learning: minimize errors between signal and output

Loss Function:
e.g. Cross-Entropy Loss

$$-\log \frac{\exp(f_\theta(x,y))}{\sum_{y' \in \mathcal{Y}} \exp(f_\theta(x,y'))}$$

"Knowledge Graph Embedding: A Survey of Approaches and Applications",
Wang et. al., IEEE TKDE, 2017.

Our work: Loss Functions

We also have studied loss functions for Knowledge Graph Embeddings:

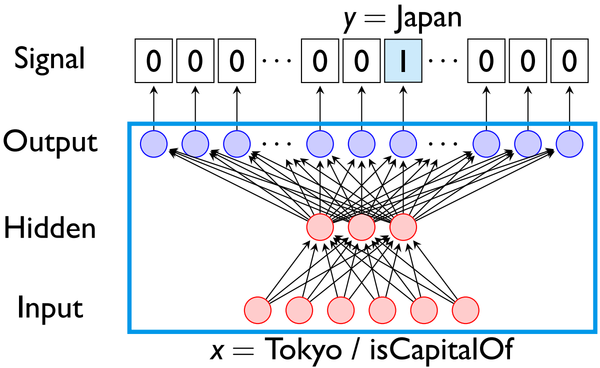
1. "Unified Interpretation of Softmax Cross-Entropy and Negative Sampling: With Case Study for Knowledge Graph Embedding",
   Hidetaka Kamigaito and Katsuhiko Hayashi, ACL, 2021.

2. "Comprehensive Analysis of Negative Sampling in Knowledge Graph Representation Learning",
   Hidetaka Kamigaito and Katsuhiko Hayashi, ICML, 2022.

For the details, refer to a tutorial by Hidetaka Kamigaito

- `https://stair.center/archives/3111`

Scoring Functions

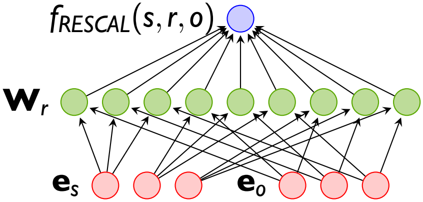


Scoring Function $f_\theta(x, y)$:
designed to measure the plausibility of triples
e.g. RESCAL model [ICML2011]

$$f_\theta(x = s/r, y = o) = \mathbf{e}_s^{\mathrm{T}} \mathbf{W}_r \mathbf{e}_o$$

“Knowledge Graph Embedding: A Survey of Approaches and Applications”,
Wang et. al., IEEE TKDE, 2017.

RESCAL Model [ICML2011]



"A Three-Way Model for Collective Learning on Multi-Relational Data", Nickel et. al., ICML, 2011.

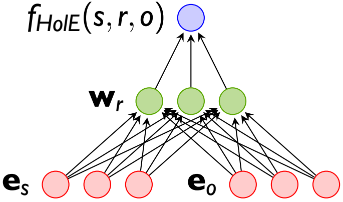
Each relation is represented as a matrix which models pairwise interactions between latent factors:

$$f_{RESCAL}(s,r,o) = \mathbf{e}_s^{\mathrm{T}} \mathbf{W}_r \mathbf{e}_o$$

RESCAL requires $O(d^2)$ parameters per relation

RESCAL is **fully expressive**: If given any ground truth, there exists an assignment of values to the embeddings of the entities and relations that accurately separates the correct triples from incorrect ones

HolE Model [AAAI2016]



“Holographic Embeddings of Knowledge Graphs”, Nickel et. al., AAAI, 2016.

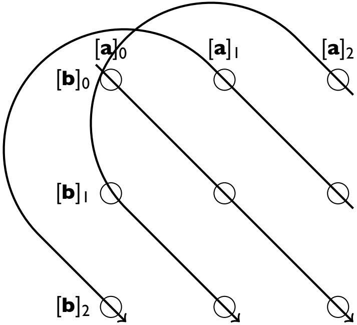
HolE requires only $O(d)$ parameters per relation

$$f_{HolE}(s,r,o) = \mathbf{w}_r^{\mathrm{T}}(\mathbf{e}_s \star \mathbf{e}_o)$$

Since circular correlation $\star$ is not commutative, i.e., $\mathbf{x} \star \mathbf{y} \neq \mathbf{y} \star \mathbf{x}$, HolE is able to handle asymmetric relations as RESCAL does (fully expressive)

HRR memory: $\mathbf{e}_o = \sum_{r'(s',o) \in \mathcal{K}} \mathbf{e}_{s'} * \mathbf{w}_{r'}$ acts as a memory $\mathbf{m}$ that stores the set of all s-r pairs for which $r'(s',o)$ is true

Circular Correlation



$$[\mathbf{a} \star \mathbf{b}]_j = \sum_{k=0}^{d-1} [\mathbf{a}]_k [\mathbf{b}]_{(k+j)}$$

$$(\text{for } j = 0, \ldots, d-1)$$

Circular correlation via the Discrete Fourier Transform can be computed in $O(d \log d)$ time

$$\begin{aligned} f_{HolE}(s,r,o) &= \mathbf{w}_r^{\mathrm{T}} (\mathbf{y}_s \star \mathbf{y}_o) \\ &= \mathbf{w}_r^{\mathrm{T}} \mathfrak{F}^{-1} (\overline{\mathfrak{F}(\mathbf{y}_s)} \odot \mathfrak{F}(\mathbf{y}_o)) \end{aligned}$$

where $\mathfrak{F}$ is the discrete Fourier transform (DFT) and $\mathfrak{F}^{-1}$ is the inverse DFT

Spectral HolE Model [ACL2017]

“On the Equivalence of Holographic and Complex Embeddings for Link Prediction”, Hayashi and Shimbo, ACL, 2017.

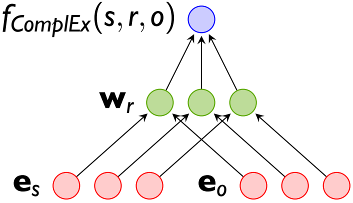
operation	time		frequency
scalar mult.	$\alpha\mathbf{x}$	$\leftrightarrow$	$\alpha\,\mathfrak{F}(\mathbf{x})$
summation	$\mathbf{x}+\mathbf{y}$	$\leftrightarrow$	$\mathfrak{F}(\mathbf{x})+\mathfrak{F}(\mathbf{y})$
flip	$\mathrm{flip}(\mathbf{x})$	$\leftrightarrow$	$\overline{\mathfrak{F}(\mathbf{x})}$
convolution	$\mathbf{x}*\mathbf{y}$	$\leftrightarrow$	$\mathfrak{F}(\mathbf{x})\odot\mathfrak{F}(\mathbf{y})$
correlation	$\mathbf{x}\star\mathbf{y}$	$\leftrightarrow$	$\overline{\mathfrak{F}(\mathbf{x})}\odot\mathfrak{F}(\mathbf{y})$
dot product	$\mathbf{x}^{\mathrm{T}}\mathbf{y}$	$=$	$\frac{1}{n}\mathfrak{F}(\mathbf{x})^{\mathrm{T}}\mathfrak{F}(\mathbf{y})$

We proposed a method how to train HolE solely in the frequency domain

$$
\begin{aligned}
f_{HolE}(s,r,o) &= \mathbf{w}_r^{\mathrm{T}}(\mathbf{y}_s\star\mathbf{y}_o)\\
&= \frac{1}{n}\omega_r^{\mathrm{T}}(\overline{\varepsilon_s}\odot\varepsilon_o)
\end{aligned}
$$

where $\omega_r=\mathfrak{F}(\mathbf{w}_r)$, $\varepsilon_s=\mathfrak{F}(\mathbf{e}_s)$ and $\varepsilon_o=\mathfrak{F}(\mathbf{e}_o)$ are d-dim. complex vectors

ComplEx Model [ICML2016]



"Complex Embeddings for Simple Link Prediction",
Trouillon et. al., ICML, 2016.

Embedding space = complex space (not real space)

$$f_{ComplEx}(s, r, o) = \Re\left(\sum_{i=0}^{d-1} [\mathbf{e}_s]_i [\mathbf{w}_r]_i \overline{[\mathbf{e}_o]_i}\right)$$

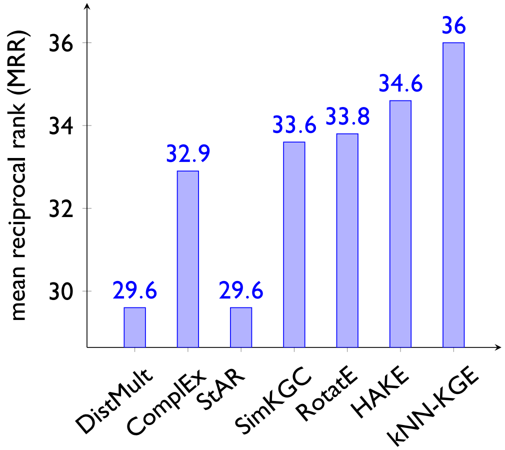
where $\Re(\cdot)$ means taking the real part of a complex value

ComplEx = HolE [Hayashi, ACL2017]

$$f_{HolE}(s, r, o) = \frac{2}{d} f_{ComplEx}(s, r, o)$$

ComplEx is Simple but Tough to Beat

KGC performance on FB15k-237



FB15k-237: KGC benchmark dataset

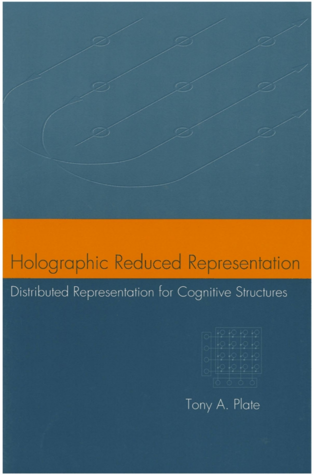
RotatE [ICLR2019] and HAKE [AAAI2020] are current SoTa models

ComplEx also achieves comparable results with StAR [TACL2021], SimKGC [ACL2022] and kNN-KGE [2023], which are KGC models based on LLMs

TODAY'S AGENDA

- ▶ Background
- ▶ Representation Learning with HRR
- ▶ Circular HRR and Binary Spatter Codes

HRR (Notify Again)



Holographic Reduced Representation

Distributed Representation for Cognitive Structures

Tony A. Plate

HRR is a method for representing compositional structure in distributed representation [IJCAI1991]

Eg.: Frame Structure “Mark eats the fish”

$$\mathbf{f} = \mathbf{eat} + \mathbf{agt}_{eat} * \mathbf{Mark} + \mathbf{obj}_{eat} * \mathbf{Fish}$$

where $*$ is an encoding operator

HRR has a decoding operator

$$\mathbf{agt}_{eat} \star \mathbf{f} \approx \mathbf{Mark}$$

Complex-valued HRR System

It might be possible to work with complex numbers and avoid convolution and Fourier transforms altogether

operation	time		frequency
scalar mult.	$\alpha \mathbf{x}$	$\leftrightarrow$	$\alpha \mathfrak{F}(\mathbf{x})$
summation	$\mathbf{x} + \mathbf{y}$	$\leftrightarrow$	$\mathfrak{F}(\mathbf{x}) + \mathfrak{F}(\mathbf{y})$
flip	$\text{flip}(\mathbf{x})$	$\leftrightarrow$	$\overline{\mathfrak{F}(\mathbf{x})}$
convolution	$\mathbf{x} * \mathbf{y}$	$\leftrightarrow$	$\mathfrak{F}(\mathbf{x}) \odot \mathfrak{F}(\mathbf{y})$
correlation	$\mathbf{x} \star \mathbf{y}$	$\leftrightarrow$	$\overline{\mathfrak{F}(\mathbf{x})} \odot \mathfrak{F}(\mathbf{y})$
dot product	$\mathbf{x}^{\mathrm{T}} \mathbf{y}$	$=$	$\frac{1}{n} \mathfrak{F}(\mathbf{x})^{\mathrm{T}} \mathfrak{F}(\mathbf{y})$

Initialization of HRR vectors:

$$\mathbf{x}' = \mathfrak{F}(\mathbf{x} = [x_0, \ldots, x_{d-1}]^{\mathrm{T}})$$

where $x_i \overset{d}{=} N(0, \frac{1}{d})$

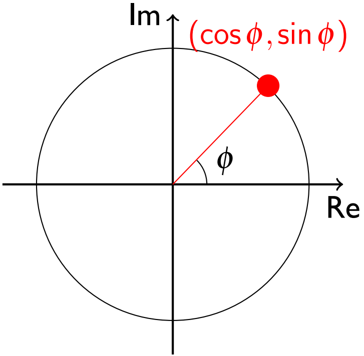
We have correspondence between operations in time and frequency domains

Circular HRR System

Tony A. Plate proposed **circular HRR system** in his thesis [1994]:

operation	
vector	$\bar{\phi} = [\phi_0, \dots, \phi_{d-1}]$ where $\phi_i \overset{d}{=} U(-\pi, \pi)$
dot product	$\bar{\phi} \cdot \bar{\theta} = \frac{1}{d} \sum_i \cos(\phi_i - \theta_i)$
binding	$\bar{\phi} \odot \bar{\theta} = [(\phi_0 + \theta_0) \mathrm{mod} 2\pi, \dots,$ $(\phi_{d-1} + \theta_{d-1}) \mathrm{mod} 2\pi]$
decoding	$-\bar{\phi} \odot \bar{\theta}$
addition	$\bar{\phi} \oplus \bar{\theta} = [\zeta_0, \dots, \zeta_{d-1}]$ where $\cos(\zeta_i) = \cos(\phi_i) + \cos(\theta_i)$ and $\sin(\zeta_i) = \sin(\phi_i) + \sin(\theta_i)$

The polar form of a complex number:



Relationship to Binary Spatter Coding

Binary spatter codes (BSC) [Kanerva,1996] are identical to the circular HRR system working on unitary vectors with phase angles quantized to 0 and π

- The two possible values $1 + 0i$ and $-1 + 0i$

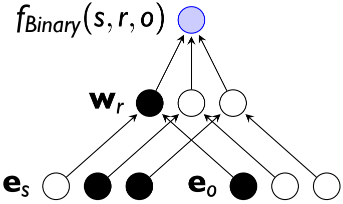
BSC use **exclusive-OR (XOR)** for binding binary vectors

- Elementwise multiplication of complex vectors whose elements can be $1 + 0i$ or $-1 + 0i$ is equivalent to XOR (more strictly, the logical complement of XOR (XNOR))

BSC use **the majority rule** for the addition of a set of vectors

- the sum of such vectors followed by elementwise quantization to $1 + 0i$ or $-1 + 0i$ is equivalent to computing the elementwise majority

Our work: Binarized Embeddings



“Binarized Knowledge Graph Embeddings”, Kishimoto et. al., ECIR, 2019.

- Scoring Function $f_{Binary}(s, r, o)$:

$$-Ham(\mathbf{e}_s, XNOR(\mathbf{e}_o, \mathbf{w}_r))$$

Other papers:

- “A Greedy Bit-flip Training Algorithm for Binarized Knowledge Graph Embeddings”, Katsuhiko Hayashi, Koki Kishimoto, Masashi Shimbo, Findings of EMNLP, 2020.

- “Binarized Embeddings for Fast, Space-Efficient Knowledge Graph Completion”, Katsuhiko Hayashi, Koki Kishimoto, Masashi Shimbo, IEEE TKDE, 2023.

Conclusion

I introduced a method using circular convolution for encoding compositional structured data into low-dimensional vector space

- Multi-label Learning with HRR
- Graph Representation Learning with HRR
- Complex/Binary-valued HRR Systems

I am now interested in designing more effective vector symbolic architechtures

I seek a research collaborator

- Please feel free to contact me: katsuhiko-h@ist.hokudai.ac.jp