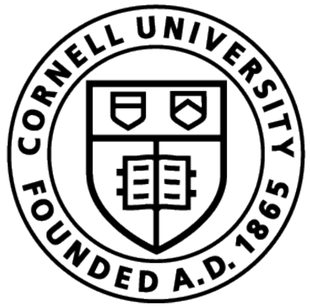
Understanding Hyperdimensional Computing for Parallel Single-Pass Learning

Tao Yu
Cornell University

Joint work with Yichi Zhang, Zhiru Zhang and Christopher De Sa



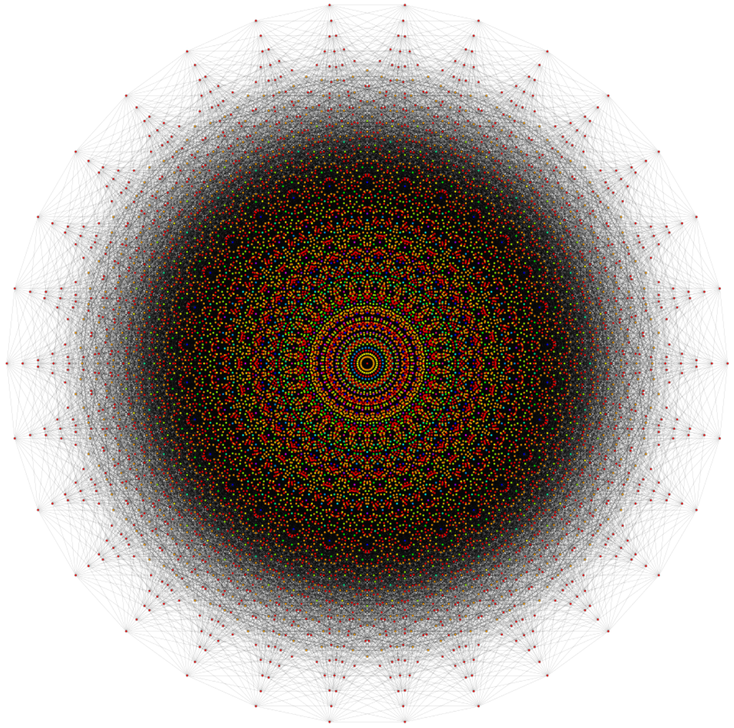
Cornell Bowers C·IS
Computer Science

Hyperdimensional Computing (HDC)

- Hypervectors = very high dimensional vectors
  - e.g. $u \in \{0,1\}^D, D = 10000$, a point in the hyper dimensional space, or a corner of a hypercube
- Operations of HDC can be highly parallel
- Robust to noises …

Given a corner u, how far is it to other corners?

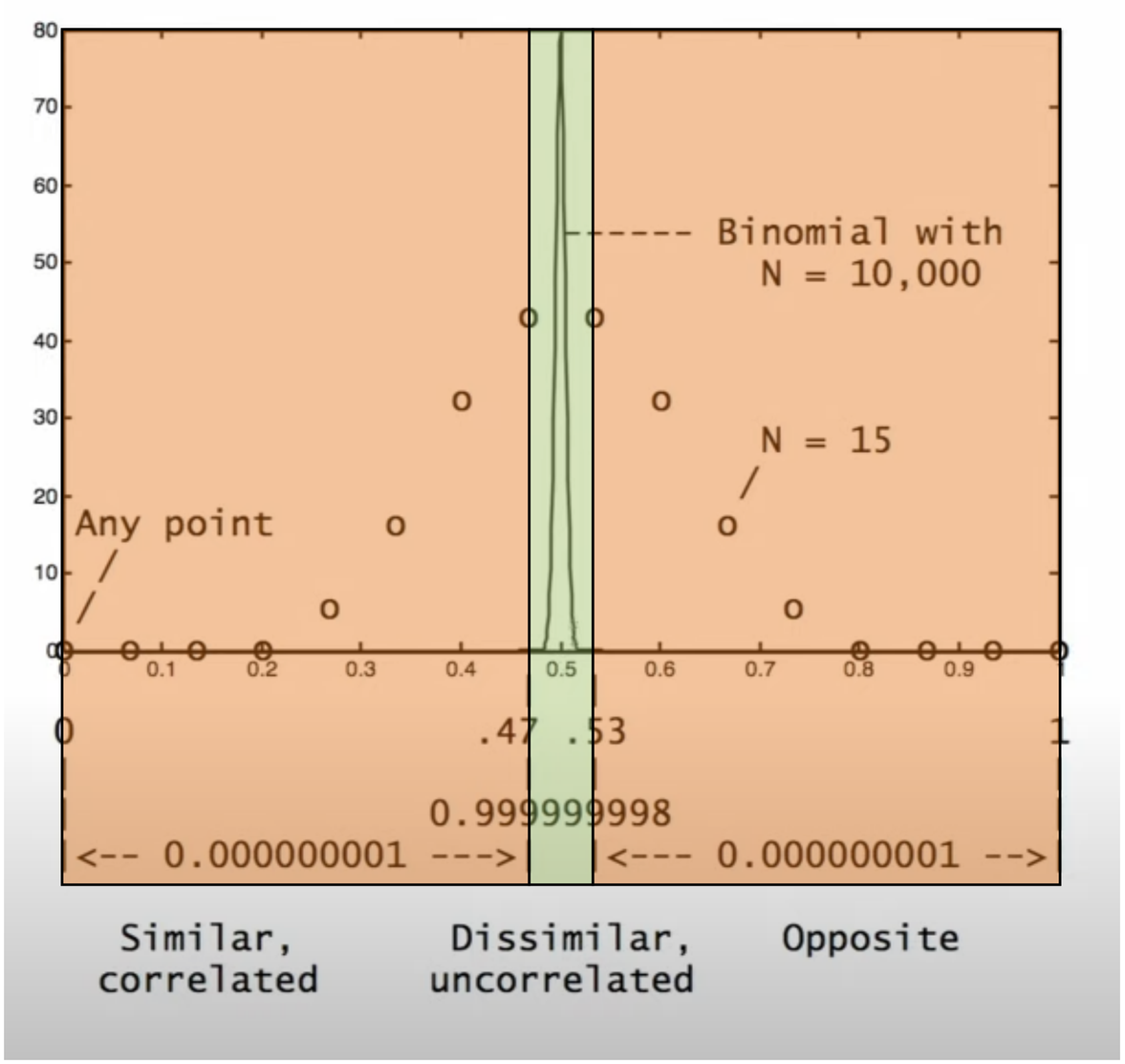
# of Corners at Hamming distance k to u: $\binom{D}{k}$



15-cube

https://en.wikipedia.org/wiki/Hypercube

Distance of a Corner to Other Corners



- $P(\frac{distance}{D} \leq 0.47) \leq$ a thousand-millionth
- $P(\frac{distance}{D} \geq 0.53) \leq$ a thousand-millionth
- A 600-bit wide “bulge” contains nearly all of the space!
- Two random vectors differ in ~5000 bits, such vectors are unrelated (orthogonal).
- Even 1/3 of the bits in a 10,000D vector are flipped, it can still be recognized, as it is closer to the original “error-free” vector than any unrelated vector.

Hyperdimensional Representations

- Hypervectors in a high-dimensional space, hyperspace
  - e.g. $\{0,1\}^D$ (BSC), $\{-1,1\}^D$ (MAP-B), $\mathbb{Z}^D$ (MAP-I), $\mathbb{R}^D$ (MAP-C), $\mathbb{C}^D$ (FHRR) …
  - Given a random hypervector v, most vectors in this hyperspace are orthogonal to v
    - Independent random hypervectors are unrelated and can naturally represent objects that are semantically separate
    - Two hypervectors u and v that have a high-enough inner-product similarity can be classified as being related with high probability.
- HDC represents data using random hypervectors and computes using a fixed set of operations: **similarity, binding, bundling, and permutation**.

- A **similarity** function $\mathcal{S}(u, v)$ measures how close/similar two hypervectors are, typically defined as an inner product function $\mathcal{S}(u, v) = \frac{1}{D}\sum_{i=1}^{D} u_i v_i$ e.g. $\{-1,1\}^D$

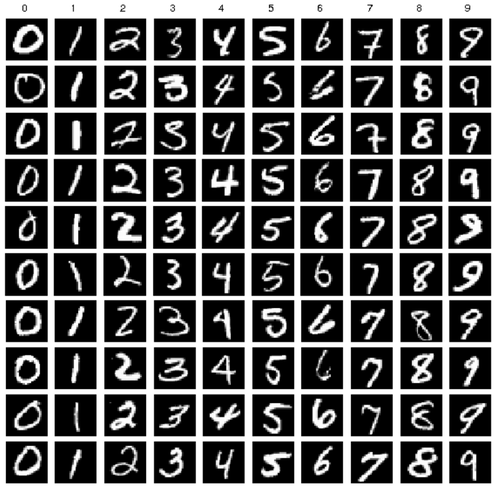
Hyperdimensional Arithmetic

- **Binding** $\otimes$ (commutative)
  - Connect a pair of hypervectors u, v into a new hypervector $u \otimes v$
  - $u \otimes v$ is non-similar to u and v
  - Similarity is preserved, i.e., $\mathcal{S}(u \otimes w, v \otimes w) = \mathcal{S}(u, v)$
  - For $\{-1,1\}^D$, $\otimes$: coordinate-wise multiplication: $(u \otimes v)_i = u_i v_i$
- **Bundling** $\oplus$
  - Aggregate a set of hypervectors and output a representative hypervector that is maximally similarly to its inputs
  - $\bigoplus_{i=1}^{m} x_i = \arg\max_g \sum_{i=1}^{m} \mathcal{S}(g, x_i)$
  - For $\{-1,1\}^D$, $\oplus$: $\bigoplus_{k=1}^{m} x_k = \text{sgn}(\sum_{k=1}^{m} x_k)$ (element-wise)
- **Permutation** Π
  - An invertible shuffling of the elements in a hypervector
  - Πu is non-similar to u
  - Useful for encoding order and position information

Example 1: Currency Retrieving

- Goal: what is the currency of a country?
  - Country: US, China
  - Currency: Dollar, CNY
- Encode:
  - Assign a random **basis hypervector** to each entity
    - US: C_1, China: C_2
    - Dollar: M_1, CNY: M_2
  - Compute a filter representative $S = (C_1 \otimes M_1) \oplus (C_2 \otimes M_2)$
- Query: what is the currency of the US?
  - Compute $R = C_1 \otimes S = C_1 \otimes C_1 \otimes M_1 \oplus C_1 \otimes C_2 \otimes M_2 = M_1 \oplus C_1 \otimes C_2 \otimes M_2$
    - Similar to M_1 and $C_1 \otimes C_2 \otimes M_2$, the latter one is not a meaningful entity and will not be similar to M_2, i.e, $R = M_1 \oplus$ noise
  - Find the currency hypervector with the **highest similarity** $\mathcal{S}(R, M)$ and return corresponding currency
  - A: Dollar

Example 2: MNIST



- Dataset: MNIST
  - Handwritten digit recognition task of 28x28 grey images
  - 60000 train samples and 10000 test samples.
  - 10 classes {0,1, …, 9}

- Encode:
  - Draw 256 random **basis hypervector**s $\{v_0, v_1, \ldots, v_{255}\}$ from $\{-1,1\}^D$ to represent pixel intensities
    - each v_i represents a pixel intensity i
  - Bind all 784 (28x28) pixels by corresponding hypervectors
    - Binding is commutative, but pixels in an image have different meaningful relative positions, each pixel hypervector is shifted before binding to preserve that position information
    - Assume the input pixel intensities are $p_0, p_1, \ldots, p_{783}$, then its encoded hypervector is $t = v_{p_0} \otimes (\Pi v_{p_1}) \otimes \cdots \otimes (\Pi^{783} v_{p_{783}})$

- Learning via bundling
  - Bundle all the hypervectors that are from the same digit to generate a representative, i.e., class vector $s_c = \bigoplus_{i|y_i=c} t_i$
  - Each training image is used only once

- Inference
  - At test time, a given test image is encoded through the same procedure to get t_{test}, then compared to each class vector s_c, outputs the class c with the **highest similarity** $\mathcal{S}(t_{\text{test}}, s_c)$.

Outline:

- Limits of HDC
  — Similarity Matrix —> Expressivity of HDC
  — Limitations due to Initialization

- Solution 1: Encoding via Random Fourier Features

- Solution 2: Group HDC/VSA

- Experiments
  — Performance
  — Circuit Depth Complexity

- Conclusion

Limits of HDC

- The dimension of the hypervectors D —> expressivity of HDC, will increasing D solve all problems?

- Observation: the pair-wise similarities of basis hypervectors matters
  - Given n basis hypervectors $v_1, v_2, \ldots, v_n \in \{-1,1\}^D$, the similarity matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the similarity of pairs of hypervectors, i.e., $\mathbf{M}_{ij} = \mathcal{S}(v_i, v_j)$
  - Similarity matrix —> expressivity of HDC
- Question: Is there any similarity matrix in a HDC model that can not be achieved no matter how large D is?
  - A simple case when there are only 3 basic entities (or basis hypervectors), i.e., $n = 3$
  - ***Lemma 1: Binary HDC at any dimension (e.g. $\{-1,1\}^D$, MAP-B) can not express the matrix:***

$$\mathbf{M}_{\text{Lemma 1}} = \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix}$$

OK, but what's the result of this?

Similarity Matrix —> Expressivity of HDC

- An example task for which whether a HDC approach can learn the Bayes-optimal classifier depends on whether it can express $\mathbf{M}_{\text{Lemma 1}}$.

- ***Lemma 2: Binary HDC at any dimension cannot learn the following task.***
  - Consider a supervised learning task with input example set $\mathcal{X} = \{0,1,2\}$, output label set $\mathcal{Y} = \mathcal{X}$, and source distribution

$$\mathcal{P}(x, y) = \begin{cases} 1/9 + 2p & x = y \\ 1/9 - p & x \neq y \end{cases}$$

    for some small positive number p.
  - A HDC can ***learn*** this task if there exists a D-dimensional encoding of $\mathcal{X}$ such that, when the bundling method is used on a training set of size N drawn from $\mathcal{P}(x, y)$, the resulting classifier is the Bayes optimal classifier with arbitrarily high probability as N increases.

- ***Lemma 3: Any HDC (e.g. MAP-C, FHRR) that can express $\mathbf{M}_{\text{Lemma 1}}$ can learn this task.***

Expressible similarity matrices —> expressivity of HDC!

Limitations due to Initialization

- Is sampling hypervectors randomly a good way?
  - In such a system, any hypervector used for an encoding (used to represent a data example) is constructed either by
    - independently sampling a binary hypervector where each entry has some probability p of being 1;
    - or permuting and/or binding some pre-sampled hypervectors $u_1, u_2, \ldots, u_K$ to get encoding $v_1, v_2, \ldots, v_n$

- Further restricts the set of similarity matrices that can be expressed in expectation, where the target matrix can actually be expressed by binary HDC.

$$\left\| \mathbb{E}[\mathbf{M}] - \begin{pmatrix} 1 & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & 1 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 1 \end{pmatrix} \right\|_F \geq \frac{\sqrt{2}}{3}$$

How to express more similarity matrices?

Solution 1: Encoding via Random Fourier Features

- More principled methods to construct hypervectors:
  - If there is a target similarity matrix $\mathbf{M}$, directly instantiate hypervectors to match it in expectation
  - ***Lemma 4: If the element-wise*** $\sin(\frac{\pi}{2}\mathbf{M})$ ***is positive semi-definite, then Algorithm 1 produces hypervectors that, in expectation, exactly achieve*** $\mathbf{M}$***, otherwise, some approximation to*** $\mathbf{M}$ ***is produced.***

- Algorithm 1 can achieve more similarity matrices than the classical procedure:
  - The similarity matrix $\mathbf{M}_{\text{fail}}$ cannot be achieved in expectation by classical HDC initialization
  - Algorithm 1 can achieve $\mathbf{M}_{\text{fail}}$ as $\sin(\frac{\pi}{2}\mathbf{M}_{\text{fail}})$ is positive semidefinite

Algorithm 1 Construct correlated hypervectors

input: similarity matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$, dimension d
let $\hat{\Sigma} = \sin(\frac{\pi}{2}\mathbf{M})$ {elementwise}
let $U\Lambda U^T = \hat{\Sigma}$ {symmetric eigendecomposition}
sample $X \in \mathbb{R}^{n \times d}$ iid unit Gaussians
return $\text{sgn}(U\Lambda_+^{1/2}X)$ {elementwise}

$$\mathbf{M}_{\text{fail}} = \begin{pmatrix} 1 & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & 1 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 1 \end{pmatrix}$$

Solution 2: Group HDC/VSA

- However, as ***Lemma 1*** shows, binary HDC has inherent limits, can't express $\mathbf{M}_{\text{Lemma 1}}$.
  - Use non-binary hyperspace, e.g., $\mathbb{R}^D$ (MAP-C), $\mathbb{C}^D$ (FHRR) to surpass the limits
  - A continuous space, requiring significant hardware complexity overhead compared to binary HDC

- We propose a new class of HDC/VSA, ***finite group VSA***, which effectively "interpolates" between them so as to bypass the similarity-representation limits of binary HDC without the need for a continuous space.
  - Hypervectors in the hyperspace G^D, where G is a finite group, theoretically extend and define similarity, binding, bundling operations accordingly
  - ***Cyclic group VSA*** with $G = \mathbb{Z}/n\mathbb{Z} = \{0,1,\ldots,n-1\}$, addition modulo n as binding $\otimes$, $n = 2$ —> binary HDC, $n \to \infty$ —> $\mathbb{C}^D$ (FHRR)

- Group VSA vs $\mathbb{C}^D$ (FHRR)
  - Any similarity matrix that can be expressed by a ***finite Abelian group*** VSA can be expressed by FHRR
  - There exists similarity matrices that can be expressed by a ***non-Abelian group*** VSA, but not by FHRR.

Learning instead of Bundling

- Train an HDC model via bundling hypervectors that are in the same class $\mathbb{T}_c = \{t_i \,|\, \text{label}(t_i) = c\}$
  - A fundamental bundling assumption: the class representative s_c is similar to each t_i. This is ***not always true***, depending on the number of vectors being bundled together
  - The class vector s_c learned from bundling will be **nearly orthogonal** to each t_i in the class and no longer be its representative as $|\mathbb{T}_c|$ increases

- Represent the class representatives as a **linear classifier**, trained with SGD
  - A linear layer of size #class $\times D$, outputs per-class similarities, this helps learning class representatives and incurs minor training cost
  - The inference cost of an HDC model remains the same as the bundling case.
  - For example, the binary HDC, the classifier executes a binarized matrix multiplication as inference, i.e., $O = X \cdot \text{sgn}(W)$, where X is the input and W is the weight matrix.

Experiment - Performance

- Datasets and tasks:
  - ISOLET, a speech recognition dataset consists of audio signals (7719 samples). The goal is predicting which letter-name was spoken.
  - UCIHAR, a human activity recognition database, consists of features collected from smartphone sensors (10299 samples). The task is predicting which type of activity a human was performing.
  - MNIST and Fashion-MNIST (Xiao et al., 2017), which are more challenging for HDC.

Table 1: Comparison on test accuracy of proposed methods to SOTA HDC† Imani et al. [2019], dynamic HDC* Chuang et al. [2020] and 1-bit RFF perceptron. Dimension of hypervectors is 10,000. 1-Epo: 1-Epoch, 10-Epo: 10-Epoch.

Dataset	ISOLET		UCIHAR		MNIST		Fashion-MNIST	
Acc(%)	**1-Epo**	**10-Epo**	**1-Epo**	**10-Epo**	**1-Epo**	**10-Epo**	**1-Epo**	**10-Epo**
Percep.	82.8	90.1	69.3	91.4	94.3	94.3	79.5	79.5
HDC†	85.6	91.5	87.3	95.7	NA	89.0	NA	NA
RFF HDC	90.6	94.4	93.8	95.7	95.4	95.4	83.4	84.0
RFF G(2^3)-VSA	93.1	94.4	95.1	95.6	96.3	95.7	85.4	**86.7**
RFF G(2^4)-VSA	**94.4**	**96.0**	**95.5**	**96.6**	**96.5**	**96.6**	**87.4**	86.5

- RFF HDC already improves non-trivially over the baseline SOTA HDC.
- Group VSA improves the model accuracy further.
- Our HDC models learned from a single pass over the data achieve high accuracy.

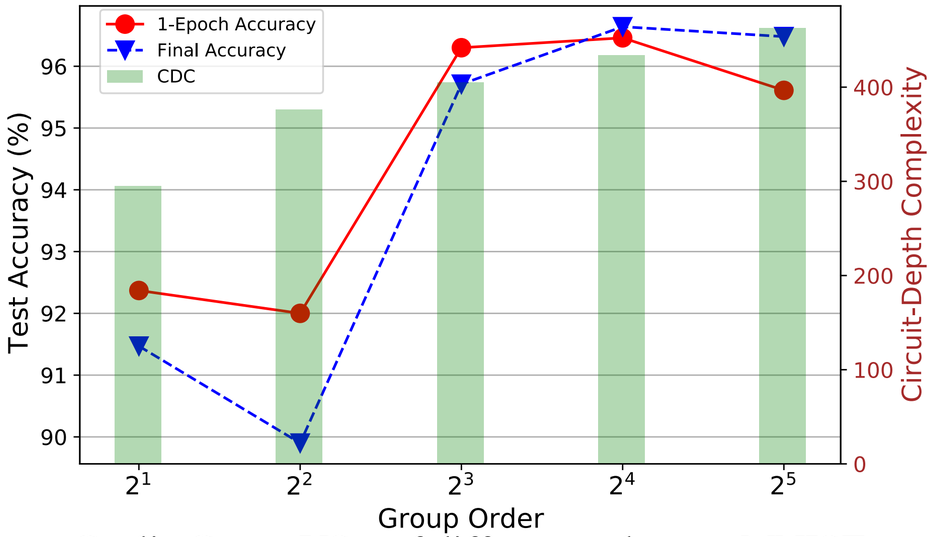
Experiment - Circuit Depth Complexity

We analyze the *circuit-depth complexity* (CDC) to quantify the potential hardware latency:

- CDC is commonly used to analyze the computational complexity of Boolean functions, defined as the length of the longest path from the input to the output (measured by the number of two-input gates along the path)
- We further assume that operations without data dependencies can be fully executed in parallel. This makes CDC independent of hardware design choices such as tiling
- N: feature vector length, e.g., 784 for an MNIST image; D: hypervector dimension.

Table 2: Analysis of circuit-depth complexity of binary HDC and 1-bit RFF perceptron.

Method	CDC
Percep.	$91 + 96 \cdot \log_2 N + \frac{3}{2} \log_2 D \cdot (1 + \log_2 D)$
HDC	$\log_2 N + 1 + \frac{3}{2} \log_2 D \cdot (1 + \log_2 D)$
G(2^n)-VSA	$3n \log_2 N + 24 \log_2 D$



Cyclic Group VSA of different orders on MNIST

- CDC on MNIST: Binary HDC 295, cyclic group $G(2^3)$ 405, 1-bit RFF perceptron 1299

Conclusion

- There is a clear connection between the class of expressible similarity matrices and the expressivity of HDC/VSA.

- This new notion of expressivity reveals the limits of HDC that computes with binary hypervectors, and meanwhile provides a hint on how we can improve it.

- The nontrivial improvement from group VSA and the proposed techniques on HDC across various benchmarks suggests that this notion paves a new way towards the future development of HDC/VSA

Thank You!