Hyperdimensional Computing and its Applications in tinyML

Ellis Weglewski

Humans

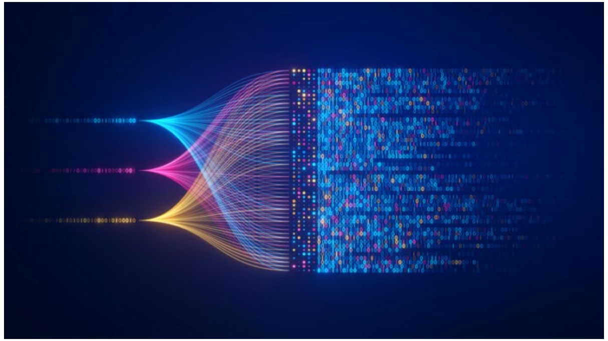
- Pattern recognizing creatures
- Excel at identifying **similar** things
- Distinguish **dissimilar** things
- Comparison

https://optimise2.assets-servd.host/chorlton-fireworks/production/images/single-firework-burst-in-sky.jpg?w=1500&q=80&fm=webp&fit=crop&crop=focalpoint&fp-x=0.5&fp-y=0.5&dm=1692611475&s=ad6c26c5ba792c4f8420f6e98065d54f
https://i.natgeofe.com/k/8fa25ea4-6409-47fb-b3cc-4af8e0dc9616/red-eyed-tree-frog-on-leaves-3-2_3x4.jpg
https://wildfoods4wildlife.com/wp-content/uploads/2017/08/dandelion-1-ADJUSTED.jpg

Contents

Machine Learning

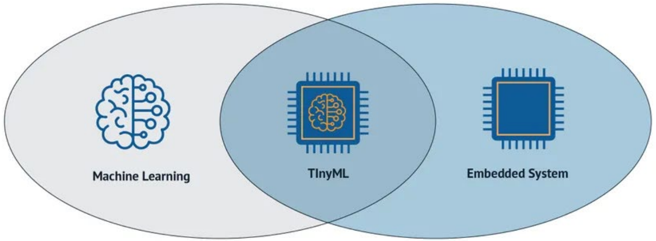
- What is it?
- Encoding data & extrapolating patterns
- Scaling complexity
- Scaling energy/time consumption



https://www.ibm.com/content/dam/connectedassets-adobe-cms/worldwide-content/stock-assets/getty/image/illustration/db/5a/dataflow001.component.xl.ts=1711716848690.jpg/content/adobe-cms/us/en/topics/machine-learning-pipeline/_jcr_content/root/leadspace

Tiny Machine Learning (tinyML)

- Embedded systems
- Tiny form factor
  - Injectables
  - Wearables
  - Implants
- Machine learning
- Held back
  - Limited Resources
  - Micro-Faults = Noise
- An approach that can deal with these problems is needed

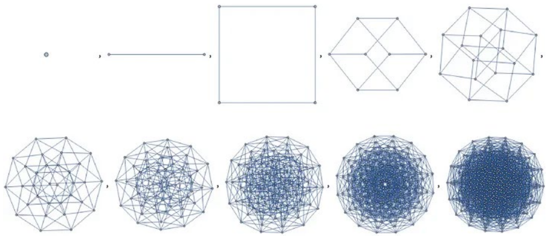


https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ2a95lQXYO0SnDy_RCgPoP95hqgeEnAACY5ntK5O8oHhLA6aA-ue_yGzzfknxnoRmb-UE&usqp=CAU

Hyperdimensional Computing (HDC)

- What is it?
- Computing Model
- Represent data as hypervectors
- Why HDC?
- Simple, Robust(Noise Resistant), Efficient, and Light
- Fits the bill

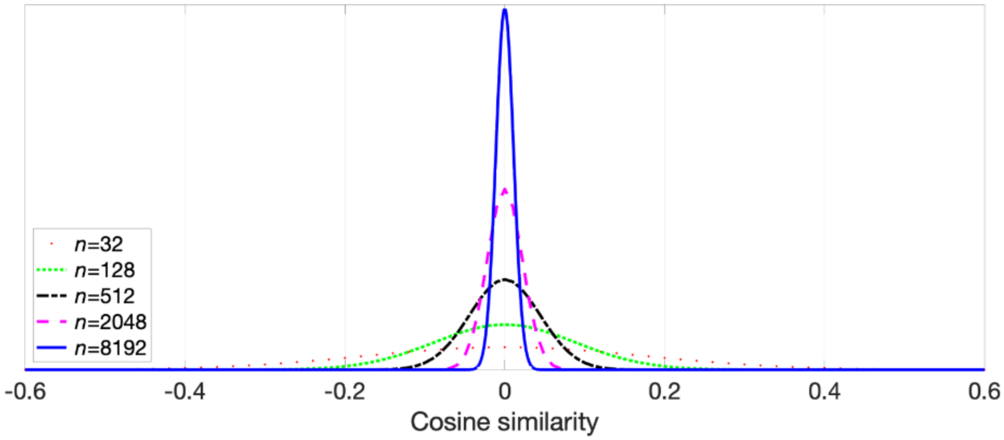
$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}.$$



https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR9xjxtvitW3nUsKnQXYTn-NYbAnxnBa2nDWMPc8j5-Ojx2mZyGwy71VbT-SSySA9Ys&usqp=CAU
https://miro.medium.com/v2/resize:fit:1400/0*DNk1CPfC9HqvpJLl

Useful Properties

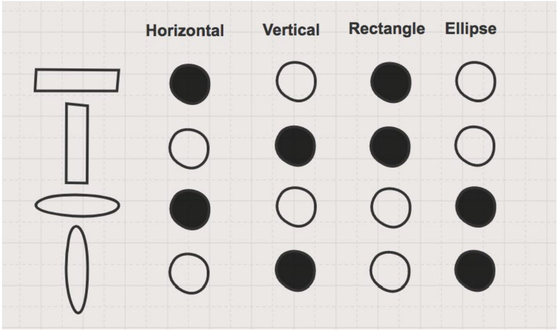
- There are reasons for hyperdimensional representation
- More capacity
- Number of orthogonal/pseudo-orthogonal vectors scales with dimensions
- Orthogonal?
- Pseudo-orthogonal?
- Cosine Similarity
- Allows for random hypervectors



https://dl.acm.org/doi/pdf/10.1145/3538531

Distributed Representation

- Data is distributed across a medium
- Vector subcomponents represent different characteristics
- Similar things will have sub-components in common



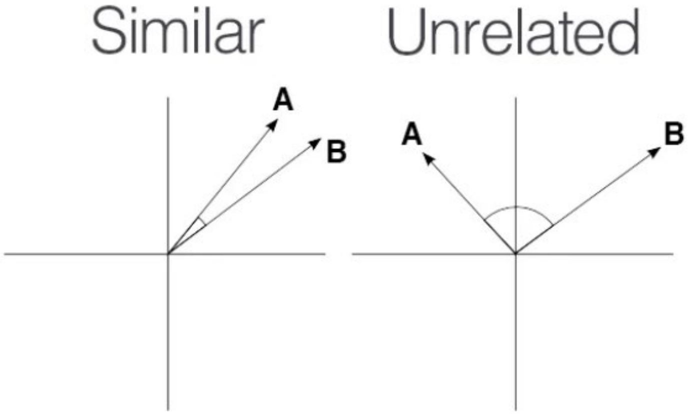
https://www.oreilly.com/content/wp-content/uploads/sites/2/2020/01/Figure2-9b757823f56b45a64820ab55a65918dd.png

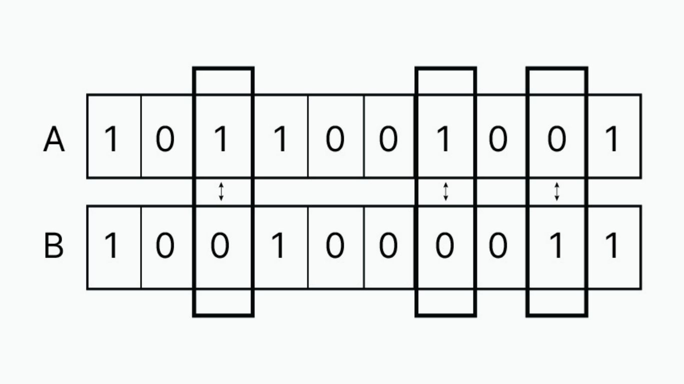
Associative Memory

- Class of memory
- Access data by content instead of address
- Allows for efficient comparison operations
- Makes it easy to find **similar** things
    - Noise resistance

Comparison Operations

- Operations to compare how similar two hypervectors are
- Cosine Similarity
- Hamming Distance





https://miro.medium.com/v2/resize:fit:915/1*dyH20eCqb6qTL-gt4nCVzQ.png

Bundling

- Higher concepts are a composition of sub-vectors
  - Red + Blue = Purple
- How can we represent a composition of sub-vectors properly?
- Addition
- Result is a hypervector that is **similar** to component sub-vectors

Binding

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{pmatrix} \odot \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{pmatrix} = \begin{pmatrix} V_1 * W_1 \\ V_2 * W_2 \\ V_3 * W_3 \\ V_4 * W_4 \end{pmatrix}$$

- We cannot rely on bundling alone, it lacks context due to associativity & commutativity
- (a+b)+(b+c) = a + b + b +c = b + a + c + b
- Context of parentheses can be established through binding
- Component-wise multiplication
- Bit-wise XOR
- Permutation
  - How do we permute?
- Result is a hypervector that is **dissimilar** to its component sub-vectors

$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$
$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$

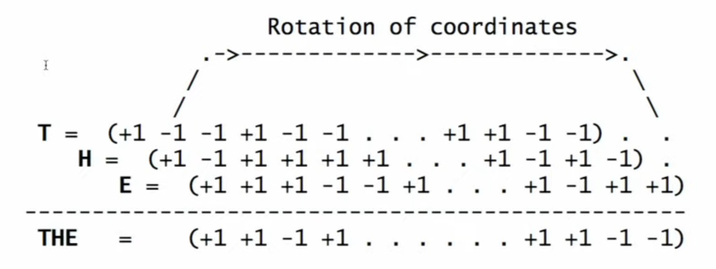
$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} a_2 \\ a_4 \\ a_1 \\ a_5 \\ a_3 \end{bmatrix}$$

https://ars.els-cdn.com/content/image/3-s2.0-B9780128114537000263-f26-07-9780128114537.jpg
https://favtutor.com/resources/images/uploads/mceu_9792096411669286980156.png
https://qph.cf2.quoracdn.net/main-qimg-f07c20a15541ad73359113cb52099153-pjlq

HDC Models

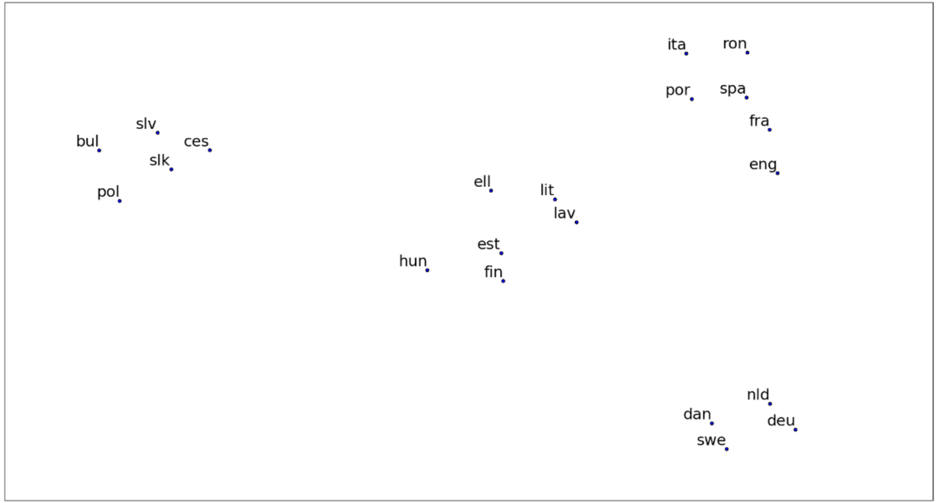
- Different hypervector models are better-suited to different problems
- Atomic hypervectors
- Binary Spatter Code (BSC)
- Multiply Add Permute (MAP)


https://chunglabblog.wordpress.com/wp-content/uploads/2021/02/12.png



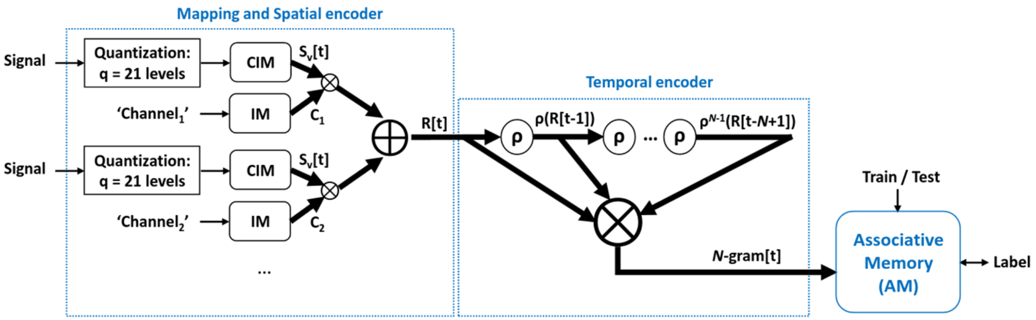
Language Identification

- Proposed by Pentti Kanerva
- MAP model
- Atomic hypervectors are the alphabet
- N-Grams
  - Tri-Grams
- Profile hypervectors
- Cosine Similarity



https://redwood.berkeley.edu/wp-content/uploads/2018/12/joshi2016language.pdf

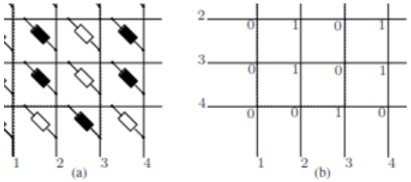
Prosthetics

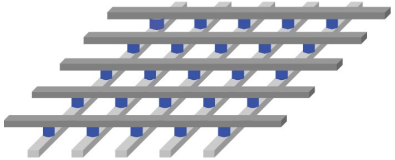


- Proposed by Abbas Rahimi
- Prosthetics still primitive
- EMG sensors
- Personalized sensor response
- BSC model
- Atomic hypervectors are channels and a seed for signal strength
    - (D/2)/(q-1)
- Record Hypervectors permuted to give temporal context
- Incoming gesture signals funneled through hamming distance for gesture

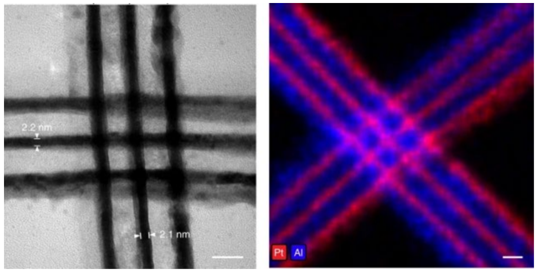
https://arxiv.org/pdf/1807.08583.pdf

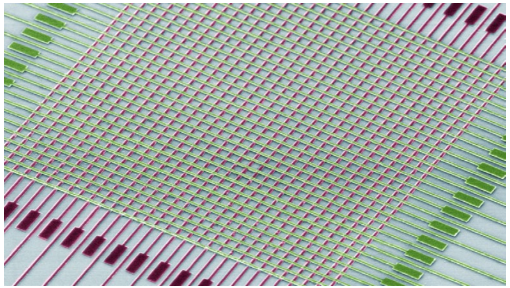
In-Memory HDC





- Time/Energy Consumption of data transfer
- Memristor
  - Capacitor
  - Inductor
  - Resistor
  - Memristor
- Crossbar Arrays
- Conductance can be changed to high/low
- Extremely energy efficient on top of negating bottleneck
- HDC can be done in memory





https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6620207
https://d3i71xaburhd42.cloudfront.net/fd3a16682c5e2be7156f1bf88804f78e89a03e8a/2-Figure1-1.png
https://www.researchgate.net/publication/328888027/figure/fig3/AS:731591059963941@1551436309448/The-2-nm-memristor-crossbar-arrays-a-A-representative-TEM-image-of-a-33-memristor.png
https://i.extremetech.com/imagery/content-types/0025SBg2aEQkwRXAtt7eU7t/hero-image.fill.size_1200x675.jpg

Conclusion

Questions?

Sources

- Denis Kleyko, Dmitri A. Rachkovskij, Evgeny Osipov, and Abbas Rahimi. 2022. A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part I: Models and Data Transformations. ACM Comput. Surv. 55, 6, Article 130 (June 2023), 40 pages.

- Denis Kleyko, Dmitri Rachkovskij, Evgeny Osipov, and Abbas Rahimi. 2023. A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part II: Applications, Cognitive Models, and Challenges. ACM Comput. Surv. 55, 9, Article 175 (September 2023), 52 pages.

- Manuel Schmuck, Luca Benini, and Abbas Rahimi. 2019. Hardware Optimizations of Dense Binary Hyperdimensional Computing: Rematerialization of Hypervectors, Binarized Bundling, and Combinational Associative Memory. J. Emerg. Technol. Comput. Syst. 15, 4, Article 32 (October 2019), 25 pages.

- Karunaratne, G., Le Gallo, M., Cherubini, G., Benini, L., Rahimi, A., & Sebastian, A. (2020). In-memory hyperdimensional computing. *Nature Electronics*, *3*(6), 327-337.

Thank You