Hyperdimensional Computing
Among the most challenging scientific questions of our time are the corresponding analytic and synthetic problems: How does the brain function? Can we design a machine which will simulate a brain?

-- Automata Studies, 1956
“The theory reported here clearly demonstrates the feasibility and fruitfulness of a quantitative statistical approach to the organization of cognitive systems. By the study of systems such as the perceptron, it is hoped that those fundamental laws of organization which are common to all information handling systems, machines and men included, may eventually be understood.” -- Frank Rosenblatt

What the Frog's Eye Tells the Frog's Brain*

J. Y. LETTVIN†, H. R. MATURANA‡, W. S. McCulloch§, Senior Member, IRE, and W. H. Pitts||

Perception, 1972, volume 1, pages 371–394

Single units and sensation: A neuron doctrine for perceptual psychology?

H B Barlow
Department of Physiology–Anatomy, University of California, Berkeley, California 94720
Received 6 December 1972

Abstract. The problem discussed is the relationship between the firing of single neurons in sensory pathways and subjectively experienced sensations. The conclusions are formulated as the following five dogmas:

1. To understand nervous function one needs to look at interactions at a cellular level, rather than either a more macroscopic or microscopic level, because behaviour depends upon the organized pattern of these intercellular interactions.
2. The sensory system is organized to achieve as complete a representation of the sensory stimulus as possible with the minimum number of active neurons.
3. Trigger features of sensory neurons are matched to redundant patterns of stimulation by experience as well as by developmental processes.
4. Perception corresponds to the activity of a small selection from the very numerous high-level neurons, each of which corresponds to a pattern of external events of the order of complexity of the events symbolized by a word.
5. High impulse frequency in such neurons corresponds to high certainty that the trigger feature is present.

The development of the concepts leading up to these speculative dogmas, their experimental basis, and some of their limitations are discussed.
NERSC (Lawrence Berkeley Lab) ~ 5 MW

Jumping spider ~ 1 fly/day

(Bair & Olshausen, 1991)
Hyperdimensional Computing

- The brain’s circuits are high-dimensional.
- Computing elements are stochastic, not deterministic.
- No two brains are alike, yet they exhibit the same behavior.
- Learns from data/example, learns by analogy, or even “one-shot.”
- Integrates signals from disparate senses.
- Allows high degree of parallelism.
Holographic Reduced Representations

Tony Plate

Vector Symbolic Architectures

Ross Gayler

Hyperdimensional Computing

Pentti Kanerva


- Everything represented as a high-dimensional vector.
- Algebra over vectors (instead of numbers).
<table>
<thead>
<tr>
<th></th>
<th>Traditional computing/AI</th>
<th>Neural nets</th>
<th>HD computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbolic computing with variables</td>
<td>✓</td>
<td>❌</td>
<td>✓</td>
</tr>
<tr>
<td>and binding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed representation</td>
<td>❌</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Learn from data</td>
<td>❌</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Robust (error-correcting)</td>
<td>❌</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>Transparent</td>
<td>✓</td>
<td>❌</td>
<td>✓</td>
</tr>
</tbody>
</table>
Holographic Reduced Representations

Tony A. Plate

\[ t = c \oplus x \]

\[ t_0 = c_0x_0 + c_2x_1 + c_1x_2 \]
\[ t_1 = c_1x_0 + c_0x_1 + c_2x_2 \]
\[ t_2 = c_2x_0 + c_1x_1 + c_0x_2 \]

\[ t_j = \sum_{k=0}^{n-1} c_k x_{j-k} \]

for \( j = 0 \) to \( n - 1 \)

(Subscripts are modulo-\( n \))
Binding via circular convolution
\[ \hat{t} = \tilde{c} \ast \tilde{x} \]

Unbinding via circular correlation
\[ \tilde{y} = \tilde{c} \bigtriangledown \hat{t} \]
\[ \tilde{y} \approx \tilde{x} \]

Composition via superposition
\[ \hat{t} = \tilde{c}_1 \ast \tilde{x}_1 + \tilde{c}_2 \ast \tilde{x}_2 \]
\[ \tilde{c}_1 \bigtriangledown \hat{t} = \tilde{c}_1 \bigtriangledown \hat{c}_1 \ast \tilde{x}_1 + \tilde{c}_1 \bigtriangledown \tilde{c}_2 \ast \tilde{x}_2 \]
\[ \approx \tilde{x}_1 + \text{noise} \]
Variable binding

‘X=a’, ‘Y=b’

\[ \tilde{t} = \tilde{x} \odot \tilde{a} + \tilde{y} \odot \tilde{b}. \]

Language

“Mark ate the fish.”

\[ \tilde{s}_1 = \text{eat} + \text{agt}_{\text{eat}} \odot \text{mark} + \text{obj}_{\text{eat}} \odot \text{the} \_ \text{fish}. \]
Computing with high-dimensional vectors

Concepts, variables, attributes are represented as high-dimensional vectors (e.g., 10,000 bits)

Three fundamental operations:
- multiplication (binding)
- addition (combining)
- permutation (sequencing)

Approximates a field

Encoding of ‘(X=A) and (Y=B) and (Z=C)’

\[
\begin{align*}
X &= 1 0 0 1 0 \ldots 0 1 \\
A &= 0 0 1 1 1 \ldots 1 1 \\
X \times A &= 1 0 1 0 1 \ldots 1 0 \\
&\Rightarrow 1 0 1 0 1 \ldots 1 0 \quad ‘(x = a)’ \\
Y &= 1 0 0 0 1 \ldots 1 0 \\
B &= 1 1 1 1 1 \ldots 0 0 \\
Y \times B &= 0 1 1 1 0 \ldots 1 0 \\
&\Rightarrow 0 1 1 1 0 \ldots 1 0 \quad ‘(y = b)’ \\
Z &= 0 1 1 0 1 \ldots 0 1 \\
C &= 1 0 0 0 1 \ldots 0 1 \\
Z \times C &= 1 1 1 0 0 \ldots 0 0 \\
&\Rightarrow 1 1 1 0 0 \ldots 0 0 \quad ‘(z = c)’ \\
H &= 2 2 3 1 1 \ldots 2 0 \\
\text{sum} &= 1 1 1 0 0 \ldots 1 0 \\
X &= 1 0 0 1 0 \ldots 0 1 \\
X \times H &= 0 1 1 1 0 \ldots 1 1 \\
&= A’ \cong A \\
\downarrow \text{Item/clean-up memory finds nearest neighbor among known vectors} \\
0 0 1 1 1 \ldots 1 1 &= A
\end{align*}
\]
Four examples

• Analogical reasoning
• Language identification via trigram statistics
• Sequence memory
• Visual working memory
Reasoning

What is the dollar of Mexico?
Analogical Mapping with Multiplication by Hypervector

What is the Dollar of Mexico?

Encoding of USA and MEXico: Name of country, Capital city, Monetary unit

USA = Nam*Us + Cap*Dc + Mon*$
MEX = Nam*Mx + Cap*Mc + Mon*P

Pairing up the two—binding

Pair = USA*MEX

Analyzing the pair

Pair = Us*Mx + Dc*Mc + $*P + noise
Literal interpretation of *Dollar of Mexico* produces nonsense:

\[
\$^\text{MEX} = \$ \times (\text{Nam}^*\text{Mx} + \text{Cap}^*\text{Mc} + \text{Mon}^*\text{P}) \\
= \$^\text{Nam}^*\text{Mx} + \$^\text{Cap}^*\text{Mc} + \$^\text{Mon}^*\text{P} \\
= \text{noise} + \text{noise} + \text{noise} \\
\text{(nothing cancels out)}
\]

However, what in Mexico corresponds to *Dollar in USA*?

\[
\$^\text{Pair} = \$ \times (\text{USA}^*\text{MEX}) \\
= \$ \times (\text{Us}^*\text{Mx} + \text{Dc}^*\text{Mc} + \$^*\text{P} + \text{noise}) \\
= \$^\text{Us}^*\text{Mx} + \$^\text{Dc}^*\text{Mc} + \$^*\$^*\text{P} + \$^*\text{noise} \\
= \text{noise} + \text{noise} + \text{P} + \text{noise} \\
= \text{P} + \text{noise} \\
\approx \text{P}
\]
Language identification from trigram statistics (Joshi, Halseth, Kanerva 2017)

Encode a trigram vector for each three-letter sequence A, B, C as

$$ABC = \rho(\rho(A)) * \rho(B) * C = \rho \rho A * \rho B * C$$

Add all trigram vectors of a text into a 10,000-D Profile Vector. For example, the text segment

“the quick brown fox jumped over ...”

gives rise to the following trigram vectors, which are added into the profile for English

$$\text{Engl} \rightarrow \text{THE} + \text{HE#} + \text{E#Q} + \text{QU} + \text{QUI} + \text{UIC} + \ldots$$
<table>
<thead>
<tr>
<th></th>
<th>ell</th>
<th>eng</th>
<th>ita</th>
<th>ces</th>
<th>est</th>
<th>spa</th>
<th>nld</th>
<th>por</th>
<th>lav</th>
<th>lit</th>
<th>ron</th>
<th>pol</th>
<th>fra</th>
<th>bul</th>
<th>deu</th>
<th>dan</th>
<th>fin</th>
<th>swe</th>
<th>slk</th>
<th>slv</th>
</tr>
</thead>
<tbody>
<tr>
<td>987</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eng</td>
<td>2 982</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ita</td>
<td></td>
<td>. 992</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ces</td>
<td>1</td>
<td></td>
<td>940</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>est</td>
<td>1</td>
<td></td>
<td></td>
<td>1 983</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>spa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>946</td>
<td>2 30</td>
<td>8 1 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nld</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>980</td>
<td>1</td>
<td></td>
<td>2 1</td>
<td></td>
<td>5 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>por</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>1 991</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lav</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 963</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 18 969</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ron</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pol</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>984</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fra</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 2</td>
<td></td>
<td>1 1 2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bul</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>984</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>deu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>985</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>974</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>993</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hun</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>989</td>
<td></td>
</tr>
<tr>
<td>swe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>974</td>
</tr>
<tr>
<td>slk</td>
<td>2</td>
<td></td>
<td>72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 18</td>
<td>6 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slv</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6 1 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LEGEND:** bul = Bulgarian, ces = Czech, dan = Danish, deu = German, ell = Greek, eng = English, est = Estonian, fin = Finnish, fra = French, hun = Hungarian, ita = Italian, lav = Latvian, lit = Lithuanian, nld = Dutch, pol = Polish, por = Portuguese, ron = Romanian, slk = Slovak, slv = Slovene, spa = Spanish, swe = Swedish.
1. **Language Vectors:** We made 10,000-D language vectors for 21 EU languages from seed vectors representing letters. Projected onto a plane, the languages cluster according to known families:

- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish

* Slovene
* Bulgarian
* Czech
* Slovak
* Polish

Clustered languages:
- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish

Clustered languages:
- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish

Clustered languages:
- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish

Clustered languages:
- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish

Clustered languages:
- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish

Clustered languages:
- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish

Clustered languages:
- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish

Clustered languages:
- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish

Clustered languages:
- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish

Clustered languages:
- Italian
- Romanian
- Portuguese
- Spanish
- French
- English
- Greek
- Lithuanian
- Latvian
- Estonian
- Finnish
- Hungarian
- Dutch
- Danish
- German
- Swedish
Reservoir computing and recurrent neural networks

Jaeger (2001), *GMD Report 148*

Maass, Natshlager & Markram (2002), *Neural Computation*
A simple working memory

\[ x(m) = Wx(m - 1) + \Phi a(m) \]

Each input gets a time tag and is added to the memory

\[ x(M) = \sum_{m=1}^{M} W^{M-m} \Phi a(m) \]

\( W \): unitary, mixing properties
\( \Phi \): random

Frady, Kleyko & Sommer (2018). *Neural Computation*
A theory for information readout

\[ p_{\text{corr}}(s(K)) = \int_{-\infty}^{\infty} \frac{dh}{\sqrt{2\pi}} e^{-\frac{1}{2}h^2} [\Phi(h + s(K))]^{D-1} \]
Image sequence storage and retrieval
Visual working memory as a superposition of ‘what’ and ‘where’ bindings
(Eric Weiss, Ph.D. thesis)

\[ m_{t+1} = m_t + r_t \odot v_t \]
Example encoding

\[ m = v_6 \odot r_{t=0} + v_5 \odot r_{t=1} + v_4 \odot r_{t=2} + \ldots \]
Example queries

Where is the ‘5’?

answer = \( \mathbf{v}_5^* \odot \mathbf{m} \)

\[
= \mathbf{v}_5^* \odot (\mathbf{v}_6 \odot \mathbf{r}_{t=0} + \mathbf{v}_5 \odot \mathbf{r}_{t=1} + \mathbf{v}_4 \odot \mathbf{r}_{t=2} + \ldots)
\]

\[
\approx \quad \text{noise} \quad + \quad \mathbf{r}_{t=1} \quad + \quad \text{noise} \quad + \quad \ldots
\]

What object is in the center?

answer = \( \mathbf{r}_{\text{center}}^* \odot \mathbf{m} \)

\[
= \mathbf{r}_{\text{center}}^* \odot (\mathbf{v}_6 \odot \mathbf{r}_{t=0} + \mathbf{v}_5 \odot \mathbf{r}_{t=1} + \mathbf{v}_4 \odot \mathbf{r}_{t=2} + \ldots)
\]

\[
\approx \quad \mathbf{v}_6 \quad + \quad \text{noise} \quad + \quad \text{noise} \quad + \quad \ldots
\]
Spatial reasoning

What is below a ‘2’ and to the left of a ‘1’?

\[ a_1 = f^{-1}(r_{\text{down}} (v_2^* \odot m)) \]
\[ a_2 = f^{-1}(r_{\text{left}} (v_1^* \odot m)) \]
\[ a_1 \odot a_2 \]

answer = \( f(a_1 \odot a_2) \odot m \)
Other efforts

• Berkeley/Stanford EE (Rabaey, Salahuddin, Mitra, Wong) - hardware implementation, cnFET’s, PCM/RRAM

• Waterloo (Eliasmith) - SPAUN

• U Maryland (Fernmuller, Aloimonos) - event-based camera robot navigation

• BMW (Mirus, Blouw, Stewart, Conradt) - vehicle position monitoring and prediction.

• VSA online seminar series: https://sites.google.com/ltu.se/vsaonline/winter-2021

• Website: https://www.hd-computing.com