Overview

- Distributed representation for natural language and analogy.
- Pentti Kanerva. Dual role of analogy in the design of a cognitive computer. In K. Holyoak, D. Gentner, and B. Kokinov, editors, *Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences*, pages 164-170. 1998.
- Pentti Kanerva. Large patterns make great symbols: An example of learning from example. NIPS*98 workshop, 1998.

Kanerva's Perspective

- Human intelligence and language are fundamentally analogical and figurative.
- Danger in relying too much on the computer metaphor.
- Growth of human mind is largely due to **analogical perceiving** and **thinking**.
- **Imitation** is a more advanced form of learning.
- Possibility of the **coevolution** of analogy and language.
- We must **put figurative meaning and analogy** at the center, to design a new kind of "cognitive" computer.

Kanerva's Perspective: Continued

- Description vs. explanation: formalisms are good for **describing**, but may not be good enough to **explain** complex things like human thought.
- The main focus is on **patterns** generated by sensory input and **new** patterns that are derived.
- "Human mind conquers the unknown by making analogies to that which is known, it understands the new in terms of the old. In so doing, it creates rich networks of mental connections and becomes robust."

Binary Spatter-Code (Kanerva 1998)

$$10110001010110101010100 0\cdots 1011010$$

- **Space of Representation:** large N-dimensional vectors (where N is very large, $1,000 < N < 10,000$). The vectors can represent:
  1. variables (**role**),
  2. values (**filler**),
  3. composed structures, and
  4. mapping between structures,

  all in the **same space**.
- **Item Memory and Clean-up Memory:** Vectors resulting from manipulations are **not exact**, and a lookup table of valid/known vectors is needed to correct errors introduced during the manipulations.

Binding and Unbinding

Operators:

- **Binding:** Basically a **bit-wise** XOR function.

$$\mathbf{x} \otimes \mathbf{a}, \text{ where}$$

$$x_i \otimes a_i, \text{ for all } i = 1..N,$$

where $\mathbf{x}$ and $\mathbf{a}$ are vectors and the meaning of the operation is the variable assignment $x = a$.

- **Unbinding:** Retrieve either the role or the filler.

$$\mathbf{x} \otimes (\mathbf{x} \otimes \mathbf{a}) = \mathbf{a} \quad \text{(retrieved the filler)}$$

$$\mathbf{a} \otimes (\mathbf{x} \otimes \mathbf{a}) = \mathbf{x} \quad \text{(retrieved the role)}$$

Binding/Unbinding Example

- Role and filler vectors:

$$\mathbf{x} = 1111011011$$

$$\mathbf{a} = 0110111001$$

- Binding

$$\mathbf{x} \otimes \mathbf{a} = 1001100010$$

- Unbinding

$$\mathbf{a} \otimes (\mathbf{x} \otimes \mathbf{a}) = 1111011011$$

You can use matlab to try this:

```
x = [ 1 1 1 1 0 1 1 0 1 1 ]
a = [ 0 1 1 0 1 1 1 0 0 1 ]
xor(x,a)
xor(a,xor(x,a))
```


Merging

- **Merging:** superimposing, also known as bundling or chunking through **normalized sum**:

$$\langle \mathbf{G} + \mathbf{H} \rangle,$$

where each resulting bit is determined by a **bit-wise majority rule for each bit position** with ties broken randomly (i.e., when the number of 0s and 1s are the same).

- Example: the relation $r(A, B)$, represented as $r1 = A$ and $r2 = B$:

$$\mathbf{R} = \langle \mathbf{r} + \mathbf{r1} \otimes \mathbf{A} + \mathbf{r2} \otimes \mathbf{B} \rangle,$$

where $\mathbf{r}$ is the name of the relation, $\mathbf{r1}$ the first role, $\mathbf{r2}$ the second role, and $\mathbf{A}$ and $\mathbf{B}$ the fillers.

Property of Merging

Property of the merging operator:

- the resulting vector is **similar** to all constituent vectors.
- Example: given 10,000-dimensional random vectors $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, and $\mathbf{r}$, we can calculate

$$\mathbf{m} = \langle \mathbf{x} + \mathbf{y} + \mathbf{z} \rangle,$$

the correlation coefficients of the resulting vector and the constituents are around 0.5:

$$corr(\mathbf{x}, \mathbf{m}) = 0.50, corr(\mathbf{y}, \mathbf{m}) = 0.48, corr(\mathbf{z}, \mathbf{m}) = 0.51.$$

while for another random vector $\mathbf{r}$ not merged in $\mathbf{m}$,

$$corr(\mathbf{r}, \mathbf{m}) = 0.01.$$

Example of Merging

$$\mathbf{x} = 1101010110$$
$$\mathbf{y} = 0000101110$$
$$\mathbf{z} = 1100010100$$
$$\mathbf{x} + \mathbf{y} + \mathbf{z} = 2201121320$$
$$\langle \mathbf{x} + \mathbf{y} + \mathbf{z} \rangle = 1100010110$$

For example, in Matlab or Octave, try:

```
x = (sign(rand(1,10000)-0.5*ones(1,10000))+ones(1,10000))/2;
y = (sign(rand(1,10000)-0.5*ones(1,10000))+ones(1,10000))/2;
m = (sign((x+y)-1.5*ones(1,10000))+ones(1,10000))/2
```

and calculate `corrcoef(x,m)`, etc.

Distributivity

- **Distributivity:** binding and unbinding operators can be distributed over the merging operator.

$$\mathbf{x} \otimes \langle \mathbf{G} + \mathbf{H} + \mathbf{I} \rangle = \langle \mathbf{x} \otimes \mathbf{G} + \mathbf{x} \otimes \mathbf{H} + \mathbf{x} \otimes \mathbf{I} \rangle$$

* This property is **key** in the analysis of BSC.

Probing

Given a representation $\mathbf{R}$ for relation $r(A, B)$, you want to find out what the first role $\mathbf{r1}$ is bound to.

$$\mathbf{R} = \langle \mathbf{r} + \mathbf{r1} \otimes \mathbf{A} + \mathbf{r2} \otimes \mathbf{B} \rangle$$

- Simple unbind $\mathbf{R}$ with $\mathbf{r1}$:

$$\mathbf{A}' = \mathbf{r1} \otimes \mathbf{R} = \mathbf{r1} \otimes \langle \mathbf{r} + \mathbf{r1} \otimes \mathbf{A} + \mathbf{r2} \otimes \mathbf{B} \rangle$$

$\mathbf{A}' = \langle \mathbf{r1} \otimes \mathbf{r} + \mathbf{r1} \otimes (\mathbf{r1} \otimes \mathbf{A}) + \mathbf{r1} \otimes (\mathbf{r2} \otimes \mathbf{B}) \rangle$, by distributivity

$\mathbf{A}' = \langle \mathbf{r1} \otimes \mathbf{r} + \mathbf{A} + \mathbf{r1} \otimes (\mathbf{r2} \otimes \mathbf{B}) \rangle$, by unbinding,

thus, $\mathbf{A}'$ is similar to $\mathbf{A}$.

- Other similar vectors $\mathbf{r1} \otimes \mathbf{r}$ and $\mathbf{r1} \otimes (\mathbf{r2} \otimes \mathbf{B})$ are not in the cleanup memory and are treated as noise.
- Among all potential filler vectors, we can find which one is most similar to $\mathbf{A}'$.

Advanced Operations

- Multiple operations can be slapped together, e.g., multiple substitutes.
- Mapping can be done in many different ways:
  - mapping between things that share structures,
  - mapping between things that share objects, etc.
- Holistic mapping and simple **analogical retrieval**.