Quantum Informatics for Cognitive, Social, and Semantic Processes: Papers from the AAAI Fall Symposium (FS-10-08)

What We Mean When We Say "What's the Dollar of Mexico?": Prototypes and Mapping in Concept Space

Pentti Kanerva
Center for the Study of Language and Information
Stanford Univerisy, Stanford, California 94305

Abstract

We assume that the brain is some kind of a computer and look at operations implied by the figurative use of language. Figurative language is pervasive, bypasses the literal meaning of what is said and is interpreted metaphorically or by analogy. Such an interpretation calls for a mapping in concept space, leading us to speculate about the nature of concept space in terms of readily computable mappings. We find that mappings of the appropriate kind are possible in high-dimensional spaces and demonstrate them with the simplest such space, namely, where the dimensions are binary. Two operations on binary vectors, one akin to addition and the other akin to multiplication, allow new representations to be composed from existing ones, and the "multiplication" operation is also suited for the mapping. The properties of high-dimensional spaces have been shown elsewhere to correspond to cognitive phenomena such as memory recall. The present ideas further suggest the suitability of high-dimensional representation for cognitive modeling.

Overview

We first look at computing from a 60-year perspective, and the challenges we face in understanding brains in computing terms as highlighted by human language. We then adjust the traditional model of computing to correspond more closely to how brains deal with information as suggested by the very size if their circuits. This leads us to regard a high-dimensional vector as the basic unit on which to compute, which in turn gets us looking into the properties of spaces with tens-of-thousands of dimensions. The exercise then becomes mathematical: how to load specific data into these vectors and how to extract data from them. We see that relatively simple operations let us do it, so long as we can deal with their approximate results. That, in turn, is possibly because of the very high dimensionality; the approximate results cannot be corrected reliably in low-dimensional spaces. The packing and unpacking operations are then viewed as mappings between points of the space—the representatives of concepts—suggesting a mechanism for analogy, with the analogy mapping being computed from examples. We then replace variables that have a privileged position in formal systems, by prototypes that are more in line with people's use of language and with the power of learning from example. After completing this picture we highlight research leading to it and speculate on directions for future research to take.

The Brain as a Computer

The electronic computer has been our premier model of the brain for over half a century, and in turn we use mental terms such as "know," "believe," and "try" to describe the behavior of computers and their programs. Assuming that the brain, and the nervous system at large, is some kind of a computer, we are lead to speculate about the nature of the computing. Something about it must be special because the behavior produced by brains is so different from what we get from our computers. One is flexible adaptive forgiving, the other rigid and brittle.

The early computers had limited memory and speed, and we thought that more of them would close the gap, but increasing them a millionfold really hasn't. We now conjecture that quantum properties of matter might allow another millionfold increase in computing power, so would that be enough? Probably not, judging by the meager returns from the first million, so long as we keep on computing the old way. There is more to the brain's computing than raw processing power, something fundamentally different in how the computing is organized.

This paper is about the nature of the representation or code in terms of which we might come to understand the brain's computing. Equipped with such an understanding we can hope one day to build computers with brainlike performance. We are kept from doing so today, more by our lack of basic understanding than by the amount of computing power we can build into silicon chips.

Although the brain's code and mode of computing cannot be resolved by direct observation, neuroscience and human behavior provide us with ample clues. The problem then becomes mathematical: of finding models that are capable of producing the behavior and also suited for building into neural-like physical systems. We will use human language as the target behavior, as it is at once easily observed, revealing, and deeply enigmatic.

Language as a Window into the Brain's Computing

Language is a late arrival in animal evolution and it is built upon a complex nervous system. Animals without a full-fledged language have memory for places, events, actions, and consequences, are capable of planning, can assume roles in a coordinated organization, learn by observing role models, and display emotions; in other words, have a rich inner life not necessarily all that different from ours. In terms of this paper, complex computing is going on prior to language. Language then gives us the ability to communicate and share aspects of our inner lives with others by means of arbitrary labels called words, and combinations thereof. That ability could be the result of no more than one or two additional functions for the brain's circuits to accomplish, one or two additional "tricks" in the computational repertoire that make the recursive composition of abstract structure possible and that allow mapping between composed structures. This view equates the innateness of our language faculty with the availability of certain computational functions. We will look at the likely nature of such functions, but first a few words about the brain's representation at large.

Large Circuits Suggest Wide Words

A prominent feature of advanced nervous systems is the size of their circuits. The number of sensory neurons is in the millions, motor control is accomplished by circuits with tens of thousands of neurons, even apparently simple cognitive functions employ circuits with hundreds of thousands of neurons, and the parts of the brain that serve as memory have billions of neurons. The immediate conclusion is that the brain's computing is some kind of mass action in which many neurons are active at once and no individual neuron is critical. Another possible conclusion is that the brain's computing cannot be analyzed at all in terms of smaller units; it cannot be compartmentalized. That would make the brain's computing inscrutable, although new models of computing could be developed from principles discussed in this paper.

A convincing case against the inscrutable brain may be possible on anatomical, physiological, evolutionary, and behavioral grounds, but that is not the point of this paper. Instead, we will proceed with a model of computing that uses very large patterns, or high-dimensional vectors—dimensionality in the tens of thousands—as basic units on which the computer operates, as that stands a chance of being a brain model, and we will look at how computing with them could bring about flexible use of language. We call it hyperdimensional computing on account of the high dimensionality, in contrast with conventional computing that is done with bytes and words of usually no more than 32 bits.

The computing architecture we envisage is not wholly different from the conventional. It could use binary words, except that now they would be 10,000 bits wide, or maybe 50,000 bits but no more than 100,000; in the discussion below we will use 10,000. We will refer to such words as 10,000-bit *patterns* or *vectors*, or *points* of a 10,000-dimensional space, and we use the distance between points as a measure of similarity of meaning. The binary representation is used here to illustrate principles that apply to computing with high-dimensional vectors in general. The vectors could be real or complex, for example, and the choice of operations for them would, of course, depend on the kinds of vectors used; or in place of vectors we could have mathematical objects of some other kind.

A word/pattern/vector/point is the least unit with meaning. The vector components take part in the meaning but in themselves are meaningless or, rather, any part of the vector—any subset of components—has the same meaning as the entire vector, only that the meaning is represented in lower resolution. The representation is highly redundant and is in radical contrast to traditional representation that eschews redundancy. Such a representation is appropriately called *holographic* or *holistic*.

Memory for Wide Words

The computer would have a *random-access memory* for storing such words, and the memory would also be addressed by them. A memory with $2^{10\,000}$ physical locations—one for each possible address—is obviously impossible, nor is one needed; sufficient capacity for storing the experiences of a lifetime is all that is needed. If a moment of experience lasting a second were represented by one such word, about three billion (2^{32}) of them would cover a century, which falls within the information capacity of the human brain. That kind of memory can be realized as an associative neural network that works approximately as follows: when the word D is stored with the word A as the address, D can later be retrieved by addressing the memory with A or with a "noisy" version of it that is similar to A. The address A is also called a memory cue.

Basic Operations for Wide Words

The computer would have circuits for a few *basic operations* that take one or several vectors as input and produce a number or a vector as output. Such operations and their use for computing is the essence of this paper. We will start with a review of how a traditional data record composed of fields is encoded holistically. This can be done with two operations, referred to descriptively as *binding* and *bundling*, notated here with $*$ and $[\ldots+\ldots]$. Binding takes two vectors and yields a third, $U * V$, that is *dissimilar* (orthogonal) to the two, and bundling takes several vectors and yields their mean vector $[U + V + \ldots + W]$ that is *maximally similar* to them. With binary vectors, pairwise Exclusive-Or (XOR, addition modulo 2) of the components can be used for binding and the majority rule can be used for bundling: the binary mean vector agrees with the majority in each of the 10,000 positions, with ties broken at random.

Composing with Wide Words

A data record consists of a set of variables (attributes, roles) and their values (fillers); for example, the variables x, y, z with values a, b, c, respectively. The variables of a traditional record are *implicit*—they are implied by their locations in the record, called fields—whereas the values are encoded explicitly in their fields. The holistic encoding is done

as follows. The variable–value pair $x = a$ is encoded by the vector $X * A$ that binds the corresponding vectors, and the entire record is encoded by the vector

$$H = [(X * A) + (Y * B) + (Z * C)]$$

which includes both the variables and the values *explicitly*, and each of them spans the entire 10,000-bit vector—there are no fields.

The operations have two very important properties: (1) binding is *invertible* (XOR is its own inverse but with other kinds of vectors the inverse is a different operation), and (2) binding (and its inverse) *distributes* over bundling:

$$\begin{aligned} D * [U + V + \ldots + W] \\ = [(D * U) + (D * V) + \ldots + (D * W)] \end{aligned}$$

(this is only approximately true for XOR if the sum is over an even number of vectors). These properties make it possible to analyze a composite vector into its constituents. For example, we can find the value of X in the bound pair $X * A$ by $X * (X * A) = (X * X) * A = A$ (because XOR is associative and its own inverse). We can also find the variable that binds A in $X * A$ by $(X * A) * A = X * (A * A) = X$. Because binding distributes over bundling, we can further find, for example, the value of X in the holistic record vector H:

$$\begin{aligned} X * H &= X * [(X * A) + (Y * B) + (Z * C)] \\ &= [(X * X * A) + (X * Y * B) + (X * Z * C)] \\ &= [A + R_1 + R_2] \\ &= A' \\ &\approx A \end{aligned}$$

Here we assume that the variables and the values are represented by approximately orthogonal vectors A, B, C, X, Y, Z, as they would be if chosen independently at random. Then also R_1 and R_2 are approximately orthogonal to each other and to the rest and so they act as random noise added to A. The system would rely on the noise tolerance of an associative memory to retrieve A when cued with A'—we assume that the original vectors have been stored in memory.

Holistic Vectors as Mappings

So far we have seen examples of vectors representing variables, values, bound pairs, and records—i.e., objects or properties of some kind. Nothing particularly new or revolutionary might be expected from the holistic representation of these things alone. The interesting possibilities arise when the vectors are used also as mappings between objects. This brings the geometric properties of the representational space to play and has no equivalent in traditional computing.

Binding with XOR provides a ready introduction to the mapping. When the variable X is bound to the value A by $X * A$, X maps A to another part of the space. Similarly, binding X to some other value A' maps that vector to another part of the space. The geometric property of interest is that the mapping preserves distance:

$$d(X * A, X * A') = d(A, A')$$

where d is the (Hamming) distance between two binary vectors. Thus, if a configuration of points—their distances from each other—represents relations between their respective objects, binding them with X moves the configuration "bodily" to a different part of the space. Thus XORing with X serves as a mapping in which the relations are maintained. This is suggestive of analogy in which a set of facts is interpreted in a new context, and what is communicated are the relations between concepts. Furthermore, a mapping vector can be *computed* from examples, which suggests a mechanism for learning from examples.

Mapping Between Analogical Structures

Learning by analogy and analogical use of language are so pervasive and natural to us that we hardly notice them. Here we are looking for a mechanism that would be equally effortless and natural. Would mapping with holistic vectors do?

Let us look at an example where countries are represented with vectors that encode their name, capital city, and monetary unit. The variables will be represented by the vectors NAM, CAP, and MON, and the holistic vectors for the United States and Mexico would be encoded by

$$\begin{aligned} \text{USTATES} &= [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})] \\ \text{MEXICO} &= [(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})] \end{aligned}$$

As before, the nine vectors that appear inside the brackets are assumed to be approximately orthogonal, and then also USTATES and MEXICO will be. If we now pair USTATES with MEXICO, we get a bundle that pairs USA with Mexico, Washington DC with Mexico City, and dollar with peso, plus noise:

$$\begin{aligned} F_{UM} &= \text{USTATES} * \text{MEXICO} \\ &= [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad + (\text{DOL} * \text{PES}) + \text{noise}] \end{aligned}$$

The derivation of F_{UM} is straight-forward using distributivity and canceling an inverse. The vector F_{UM} can now be used for mapping, to find, for example, what in Mexico corresponds to our dollar. We get

$$\begin{aligned} \text{DOL} * F_{UM} &= \text{DOL} * [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad + (\text{DOL} * \text{PES}) + \text{noise}] \\ &= [(\text{DOL} * \text{USA} * \text{MEX}) \\ &\quad + (\text{DOL} * \text{WDC} * \text{MXC}) \\ &\quad + (\text{DOL} * \text{DOL} * \text{PES}) + (\text{DOL} * \text{noise})] \\ &= [\text{noise}_1 + \text{noise}_2 + \text{PES} + \text{noise}_3] \\ &= [\text{PES} + \text{noise}_4] \\ &\approx \text{PES} \end{aligned}$$

The thing to note is that the mapping vector F_{UM} is a simple function of two other vectors, and mapping with it is likewise done with a simple function, all of it made possible by the geometry of high-dimensional space and the operations it allows.

From Variables to Prototypes

There is something peculiar about the mapping. Although the abstract notions of variable (i.e., name, capital, monetary unit) are represented in the vectors USTATES and MEXICO, they play no role in the mapping vector F_{UM}—they merely contribute to the noise term. The mapping vector has the same form as the holistic record H introduced earlier, except that in place of abstract variables we have an exemplar: the particulars for the United States, USA, WDC, and DOL, play the role of the variables in terms of which the Mexican data are interpreted. This agrees with our common use of language when we refer to the peso as the Mexican dollar, for example. The "Mexican dollar" is immediately understood even if there is no such a thing, literally speaking, and we know that it does not exist.

Let us look at another example of mapping, starting with Sweden:

$$\text{SWEDEN} \quad = \quad [(\text{NAM} * \text{SWE}) + (\text{CAP} * \text{STO}) + (\text{MON} * \text{KRO})]$$

The mapping

$$F_{SU} = \text{SWEDEN} * \text{USTATES}$$

then "interprets" the United States in terms of Sweden. When it is paired with F_{UM} that interprets Mexico in terms of the United States, we get the mapping

$$\begin{aligned} F_{SU} * F_{UM} &= (\text{SWEDEN} * \text{USTATES}) * (\text{USTATES} * \text{MEXICO}) \\ &= \text{SWEDEN} * \text{MEXICO} \\ &= F_{SM} \end{aligned}$$

that interprets Mexico in terms of Sweden. This example resembles translating a text from one language to another through a third, from Swedish to English to Spanish, and suggests the idea that different parts of a concept space can contain similar structures, and that traversing between the structures is by relatively straight-forward mapping. Or perhaps that the readily available mapping operations determine the kinds of concept spaces we can build and make use of. To go a step further with speculation, the emergence of such mapping functions could have lead to the development of human language. Language operates with labels that are largely arbitrary, and with structures built of labels, yet refer to and evoke in us images and experiences of real things in the world. This requires efficient mapping between superficial language and the inner life, a mapping that is not found in animals at large. In this paper we argue that the properties of high-dimensional spaces make such mapping functions possible.

Let us look at one more example of a computed mapping without referring to variables, this one in the context of an IQ test: "United States is to Mexico as Dollar is to what?" It is usually displayed as

United States : Mexico : : Dollar : ?

Knowing the money of each country gives us

Peso : Mexico : : Dollar : United States

From this we can compute the sought-after mapping of Dollar. First, some function F maps DOL to USTATES, $F *$ DOL $=$ USTATES, and the *same* function maps PES to MEXICO, $F *$ PES $=$ MEXICO; we will ignore here the structure encoded into USTATES and MEXICO above. By solving the two for F we get that

$$\text{USTATES} * \text{DOL} = \text{MEXICO} * \text{PES}$$

"Multiplying" both sides by (the inverse of) MEXICO then gives us

$$\text{MEXICO} * \text{USTATES} * \text{DOL} = \text{PES}$$

In other words, MEXICO $*$ USTATES maps DOL to PES and solves the IQ puzzle.

Discussion

The notion that the most significant computing by brains takes place far below the language level has been advocated forcefully by Hofstadter (1985). His work and that of his research group has focused on the central role of analogy in human mind and language, and on its illusiveness and resistance to capture in cognitive models (Hofstadter 1995, Hofstadter et al. 1995, Mitchell 1993). Analogy at large is an important topic in cognitive science (e.g., Gentner, Holyoak, & Kokinov 2001), and models of mapping implied by it include ACME (Holyoak & Thagard (1989) and MAC/FAC (Gentner and Forbus 1991). The contrast between the literal meaning of what is said and the intended massage that is understood, is highlighted in metaphor (Lakoff & Johnson 1980). The literal image created by the words is intended to be transferred to and interpreted in a new context, exemplifying the mind's reliance on prototypes: the literal meaning provides the prototype. When the mind expands its scope in this way it creates an incredibly rich web of associations and meaning.

The above models of analogy do not make significant use of the properties of high-dimensional space. A major move in that direction was made in 1994 in Plate's PhD thesis on Holographic Reduced Representation (HRR). It contains estimates of analogical similarity based on high-dimensional holistic vectors, and it includes mapping with holistic vectors (Plate 2003). Plate discusses two kinds of HRR vectors, real and complex. Their components are independent and identically distributed (i. i. d). The binding operator $*$ is circular convolution for real vectors and pairwise multiplication of the components for complex vectors. The bundling operator $[\ldots + \ldots]$ is a vector sum that is normalized. Unlike XOR that is a self-inverse, the binding operator and its inverse are different in both real and complex HRR.

The Binary Spatter Code (Kanerva 1996) has been used in this paper to demonstrate the principles of HRR. It is equivalent to a special case of the complex HRR, namely, when the complex components of the vector are restricted to the values 1 and -1. By mapping the binary 0 to the "complex" 1 and the binary 1 to the "complex" -1, the XOR becomes ordinary multiplication and the majority rule becomes the sign function of the vector sum. The Multiply–Add–Permute (MAP) model of Gayler (1998) works with real vectors and binds two vectors with the pairwise product of their components. In all these models, binding of two vectors is done by reducing their outer product matrix, and bundling is done by superposition, and the idea is to keep the dimensionality constant. Binding with the outer product

but without the reduction appears in the work of Smolensky (1990). Aerts, Czachor, and De Moor (2006, 2009) describe Geometric Algebra analogs of Holographic Reduced Representation and its kin and make a significant addition to the repertoire of mathematical spaces with which to model concepts and cognition.

So far in this paper we have dealt with the properties of high-dimensional spaces as to their suitability for composition and mapping of representations. We should also point to their suitability for learning from data. The resurgence of interest in artificial neural networks and parallel distributed processing in the 1980s was primarily about their ability to learn from data, vividly exemplified in NETtalk (Sejnowski & Rosenberg 1986) and the learning of past tenses of English verbs (Rumelhart & McClelland 1986). Later developments include Latent Semantic Analysis (Landauer & Dumais 1997) and Random Indexing (Kanerva, Kristoferson & Holst 2000), with possible applications in text search such as provided in Google (Cohen et al. 2010).

The above "neural-net" models include only superposition: the representations are vector sums, and when they are accumulated from text and refer to words, as in latent semantic analysis, they are said to capture "bag-of-word" semantics. We are beginning to see models that include these mappings in their semantic vectors (Jones & Mewhort 2007), and then the semantic vectors can be interpreted not only based on distances between them but also based on the mappings; a mapping allows us to identify semantic vectors that encode the same or similar structure but involve different sets of objects. This opens up the possibility of learning the underlying structure of data, for example, learning to produce grammatical sentences. As a long-term goal, we hope to capture important aspects of the brain's computing in an architecture for hyperdimensional computing (Kanerva 2009). It calls for treating concepts as entities in an abstract space (Gärdenfors 2000) with their relations expressed in its geometry (Widdows 2004).

Acknowledgment I thank AAAI for providing a forum for ideas about computing that relate human intelligence to the properties of abstract mathematical spaces and to the kinds of computing operations they make possible.

References

Aerts, D., Czachor, M. and De Moor, B. 2006. On geometric algebra representation of binary spatter codes. Archive link: http://arxiv.org/abs/cs/0610075

Aerts, D., Czachor, M. and De Moor, B. 2009. Geometric analogue of holographic reduced representation. *Journal of Mathematical Psychology* 53(5):389–398. Archive link: http://arxiv.org/abs/0710.2611

Cohen, T., Widdows, D., Schvaneveldt, R. W. and Rindflesch, T. C. 2010. Logical leaps and quantum connectives: Forging paths through predication space. This volume.

Gärdenfors, P. 2000. *Conceptual Spaces: The Geometry of Thought.* Cambridge, Mass.: MIT Press.

Gayler, R. W. 1998. Multiplicative binding, representation operators, and analogy. Poster abstract. In K. Holyoak, D. Gentner, and B. Kokinov eds. *Advances in analogy research*, 409. Sofia: New Bulgarian University. Full poster http://cogprints.org/502/.

Gentner, D. and Forbus, K. D. 1991. MAC/FAC: A model of similarity-based access and mapping. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, 504–509. Hillsdale, NJ: Lawrence Erlbaum Associates.

Gentner, D., Holyoak, K. J. and Kokinov, B. eds. 2001. *The Analogical Mind: Perspectives from Cognitive Science*. Cambridge, Mass.: MIT Press.

Hofstadter, D. R. 1985. Waking up from the Boolean dream, or, Subcognition as computation. In Hofstadter, D. R. *Metamagical Themas: Questions of the Essence of Mind and Pattern*, 631–665. New York: Basic Books.

Hofstadter, D. R. 1995. Speechstuff and thoughtstuff: Musings on the resonances created by words and phrases via the subliminal perception of their buried parts. In Sture Allén, ed. *Of Thoughts and Words: The Relation Between Language and Mind. Proceedings of the Nobel Symposium 92*, 217–267. London: Imperial College Press/World Scientific.

Hofstadter, D. and Fluid Analogies Research Group. 1995, *Fluid Concepts and Creative Analogies: Computational Models of the Fundamental Mechanisms of Thought*. New York: Basic Books.

Holyoak, K. J. and Thagard, P. 1989. Analogical mapping by constraint satisfaction. *Cognitive Science* 13:295–355.

Jones, M. N. and Mewhort, D. J. K. 2007. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review* 114(1):1–37.

Kanerva, P. 1996. Binary spatter-coding of ordered K-tuples. In C. von der Malsburg, W. von Seelen, J. C. Vorbruggen and B. Sendhoff, eds. *Artificial Neural Networks—ICANN 96 Proceedings. Lecture Notes in Computer Science* 1112:869–873. Berlin: Springer.

Kanerva, P. 2009. Hyperdimensional Computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation* 1(2):139–159.

Kanerva, P., Kristoferson, J. and Holst, A. 2000. Random Indexing of text samples for Latent Semantic Analysis. Poster abstract in Gleitman, L. R. and Josh, A. K. eds. *Proceedings of 22nd Annual Conference of the Cognitive Science Society,* p. 1036. Mahwah, NJ: Erlbaum.

Lakoff, G. and Johnson, M. 1980. *Metaphors We Live By*. Chicago: University of Chicago Press.

Landauer T. and Dumais S. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review* 104(2):211–240.

Mitchell, M. 1993. *Analogy-Making as Perception*. Cambridge, Mass.: MIT Press.

Plate, T. A. 2003. *Holographic Reduced Representation: Distributed Representation for Cognitive Structures*. Stanford, Calif.: CSLI Publications.

Rumelhart, D. E. and McClelland, J. L. 1986. On learning the past tenses of English verbs. In Rumelhart, D., McClelland, J. & the PDP Research Group, eds. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* vol. II, 216–271. Cambridge, Mass.: MIT Press.

Sejnowski, T. J. and Rosenberg, C. R. 1986. NETtalk: A parallel network that learns to read aloud. Report JHU/EECS-86/01, Johns Hopkins University Electrical Engineering and Computer Science.

Smolensky P. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence* 46(12):159–216.

Widdows, D. 2004. *Geometry and Meaning.* Stanford, Calif.: CSLI Publications.