Recurrent Representation Reinterpreted

David Landy
Indiana University
Bloomington, Indiana, 47401
dlandy@indiana.edu

Abstract

Unlike many classical systems, what constitutes the "representations" in a neural network is not always explicit. In many analyses, the contents of the hidden layer are taken to be the representations. From this perspective, the representation of simple recurrent networks (SRNs) are context-sensitive and not generally compositional.

However, in this paper, an alternative analysis of the way SRNs "represent" is proposed that leads to a different conclusion. It is shown that if an SRN's representation of some input is taken to be a function specifying the network's dispositional response to that input, then SRNs are in fact formally compositional. This analysis of representation is defended as both natural and formally valid. Implications for the relationship between compositionality and systematicity are then explored. It is concluded that, surprisingly, compositionality does not play a large part in explaining systematicity.

Background

Since this paper is about how the representations in an simple recurrent network (SRN) (Elman 1990) can be compositional, a large part of the paper will be spent clarifying what "the representations" really are. Unlike most classical models, where the representational scheme is preselected, no distinct aspect of a network is definitionally "the representation". The properties of the network's representation may depend on the analysis used to extract the representational scheme. Under a common interpretation, the network's representation of some input is taken to be the vector of values on the hidden layer immediately after processing that input (Elman 1990; 1995; Tabor & Tanenhaus 1999; Rodriguez, Wiles, & Elman 1999). Types are associated with regions of hidden state space in which tokens typically appear. Because inputs prior to the most recently received are stored only implicitly (in the ways a current token deviate from the most usual tokens), there is no clear way to compose tokens, and therefore no way to evaluate whether or how they may be compositional. In fact, the interesting properties of SRNs are often attributed precisely to this context-sensitivity in their static representations. Here I begin to formalize the intuition that representations may encompass not just passive structures but also active processes, by taking as the representation of an input the network's *dispositional response* to that input. That is, the representation is just the ways that a network behaves when confronted with an input. To my surprise, by following this intuition I find that SRNs are both compositional and context-independent.

Copyright © 2004, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

There is a consensus among a wide variety of cognitive scientists that thought is generally compositional (Horwich 1997; Fodor & Lepore 2002; Aydede 1997), and that furthermore, it is largely this compositionality which accounts for the many systematic aspects of human behavior. There is much less consensus about just what properties the terms 'compositionality' and 'systematicity' are intended to pick out. Broadly speaking, compositionality seems to be that property of classical formal systems (and to some extent of natural languages) that the semantic value or 'meaning' of a compound statement is built out of the meanings of its parts. That is, in order to determine the semantic value of the compound expression $x * y$, it suffices to know that x has the value 5, and that y has the value 7; one does not need to know the exact form of the subexpressions. Formally (following Zadrozny 1994), this just means that there is a homomorphism from some syntax S to a set of representations R (where S and R have arbitrary closed composition operators $\bullet$ and $\circ$). That is, there should be a function $r : S \longrightarrow R$ which satisfies

$$r_{s \bullet t} = r_s \circ r_t$$

I will call such a function a compositional representation scheme[1].

The term 'systematicity' has been used to denote several different kinds of regularities; this makes giving a general or formal definition impossible. Later, I will discuss several of these regularities, in particular regularities in the the ways that intelligent agents learn new words, represent novel arrangements of known words, and organize their representational composites. For now, 'systematicity' will be used

[1] In order for r to be a representation scheme, it should also retain sufficient information about the inputs to discriminate all the system's responses. For instance, a function which maps everything to an identity is a homomorphism, but not a representation. It is therefore useful to add the further requirement that there be a regular function from representations to the outputs of the system, so that the representations can mediate between inputs and outputs, in order for a function to denote a representation scheme.

to refer to any of this broad group of phenomena. Compositionality is often taken to guarantee, or at least encourage, all of these regularities: As Fodor 1987 has put it "OK, so here's the argument: Linguistic capacities are systematic, and that's because sentences have constituent structure." But just why and how compositional representations are supposed to explain systematicity is left unclear. The results presented here indicate that in fact these two properties may come apart rather easily; it may therefore be useful to look for additional constraints or factors which have typically been associated with compositionality.

Compositional Representations in Simple Recurrent Networks

Consider a typical SRN, with input weights W_{in}, output weights W_{out}, and recurrent connections in the hidden layer with weight matrix C, and a hidden-layer vector space H. Let S denote the closure of the set of legal inputs to the network (the network's lexicon) under concatenation ($\bullet$): if the network's inputs include 'John' (henceforth J), 'Loves' (L), and 'Mary' (M), for instance, then S includes each of these and also compound phrases such as 'John loves', 'John loves Mary', 'John John Mary John', and so on[2]. Call the collection of possible output vectors O. Label the computation by which the SRN converts input sequences of S into responses in O as i, so that for $s \in S \setminus \epsilon, i(s) \in O$ is the output that the network produces given the sequence of inputs s. The structure of the SRN mediates between the input sequences and the produced output. An alternative to taking a simple recurrent network's representation of input word w to be the regions of hidden weight space that usually result from w is to consider the representation to be the *action* that the network takes in response to w. In particular, I consider the representation of phrase s to be just that function which describes how input s transforms the current state.

Where $\overrightarrow{t} \in S$ is a single input vector (corresponding to a 'word'), consider the function $r' : S \times H \longrightarrow H$, defined as

$$r'(s, \vec{h}) = \begin{cases} \vec{h} & \text{if } s = \epsilon \\ r'(u, \overrightarrow{Sigmoid}(C \cdot \vec{h} + W_{in} \cdot \vec{t}) & \text{if } s = \vec{t}.u \end{cases}$$

The meaning of r' is quite straightforward: given an initial vector of context layer values, $\vec{h}$, and an input sequence s, $r'(s, \vec{h})$ is equal to the hidden-layer vector after s is processed through it. Next define r to be the curry of r' over s, so that $r(s) = r'(s, \vec{h})$. $r(s)$ then denotes a *function* from H to H. For convenience, $r(s)$ can be rewritten r_s. Then r_s is the function which reflects the dispositional response of the Elman network to input s under arbitrary initial context. Let $R = \{r\}_{s \in S}$, so that $r : S \longrightarrow R$. Then the response of the system is easily generated from $r_s(\vec{h})$ by the function

$$response : R \longrightarrow O, response(X) = \overrightarrow{Sigmoid}(W_{out} \cdot X(\vec{0}))$$

[2]For convenience, I will augment the lexicon with an empty input ϵ. This null value is not an all-zero input, but a non-input. Processing an ϵ corresponds to just leaving the network alone. So $\forall s \in S, s \bullet \epsilon = s$.

r_s is a *representation* function in the sense that it is a naturally defined internal disposition that leads directly to the system's response for the input s. Furthermore, these functions are classically compositional in that r is a homomorphism from S to R. For any single input, $a = a \bullet \epsilon$, note that the system's representation of that word is just the function $r_a(\vec{h}) = \overrightarrow{Sigmoid}(C \cdot \vec{h} + W_{in} \cdot \vec{t})$. Consequently, $r_{a \bullet s} = r_a \circ r_s$ (where a is a single input word). If $\vec{h} = \vec{0}$, then $r'(s, \vec{h})$ is the hidden layer after processing s from a cleared context. Since for any given pair of sequences s, t and a single input word a, it is clear that

$$\begin{aligned} r_s \circ r_{a \bullet t} &= r_s \circ (r_a \circ r_t) \\ &= (r_s \circ r_a) \circ r_t \\ &= r_{s \bullet a} \circ r_t \end{aligned}$$

we know by induction that $r_s \circ r_t = r_{s \bullet t}$, and therefore that the R under function composition is homomorphic to S under concatenation. So R is a formally compositional representation scheme defined by the weights of the SRN.

Representation as Disposition Function

It has been shown that the set of functions R is a formally compositional set defined implicitly by an SRN. In order to defend a claim that SRNs are themselves compositional, it is necessary to argue further that R is a good candidate for the role of the representational scheme of an SRN. I do not intend to enter an already crowded debate in cognitive science by trying to provide a general definition of what counts as an agent 'representing' (see, for example, Van Gelder, 1995 or Haugeland, 1985). Instead, I will point to some commonly repeated intuitions about representation, and show where functional representations do or do not embody these themes. I will focus on comparing and contrasting my function-oriented notion of representations with the usual hidden layer value interpretation. On the whole, disposition functions match prior intuitions quite well.

Representations are usually divided into tokens and types; types encode system knowledge about a world property, while the deployment of that knowledge is called a token. The typical rendition of representation in (trained) SRNs takes the token to be the value of the hidden-layer vector after receiving a particular input word. Because prior context may differ, different tokens representing a particular input typically have similar but not identical values; the representation type is identified with the general region of the state-space where tokens of that input typically appear. Hierarchical clustering is used to group similar vectors into representation types, and to illustrate similarity among different types.

Though the two interpretations seem quite different, the functions advocated here are really a quite natural extention of the usual definition of representation. Any function can be viewed as an infinite collection of input-output pairs; the system's representation type for some input M is exactly the collection of all possible pairs; the usual method considers the representation type to be a sample of the range of this function. Any particular token in the clustering view is just

the second half of the pair which the functional interpretation takes to be the representational unit. So by simply extending the usual method to include all possible, rather than typical, previous input sequences (and therefore all achievable hidden-layer states), and by including the input value in the representation, one can move from the usual notion of representation to the compositional functional notion. The practical techniques which are typically used to define and characterize representations, such as hierarchical clustering and perhaps principle components analysis should be adaptable to a functional definition of representation. Since the functional approach folds input values into the representation tokens, it is natural to distinguish between representations of individual words and whole phrases, and to compare them directly. Though such analysis has not yet been performed, it may suggest more novel analysis techniques.

A token pair describes which particular action is taken to generate an output and therefore characterizes precisely the state of affairs which mediates between inputs and outputs in a system, which (on some accounts) is just what representations are supposed to do (Markman & Dietrich 2000). Furthermore, taking as the representation the full range of potential responses of the network to an input matches an intuition that processes, and not just structure, engage in representing, and that representations in neural networks are therefore active and not passive entities. Taking the action to be an application of the dispositional response function also allows us to consider the representation properties more formally.

A system's dispositional response is a concise way of characterizing what the system knows about some input; this knowledge is what declarative representations are intended to capture. Rather than reflecting the knowledge encoded in the weights, these tokens are directly built out of the knowledge which the network employs in parsing a sequence in a particular context. This knowledge is quite naturally derived from the structure of the network; they do not employ discontinuous conditionals, or any other complicated algorithm, but instead simply multiply and add vectors (and apply the sigmoid squashing). Thus, no explicit decision is ever made in their execution. Furthermore, composite representations are derived (in the strong sense demanded by Fodor & Lepore, 2002) from their constituent elements.

Since the functions in R represent *types*, any particular application of the function is a tokening of that type. A representation token is not therefore a member or part of the enduring state of the system, but is instead something the system does, in virtue of its knowledge about some particular input. Tokening a representation is applying an action-type: the hidden-layer representation after hearing JLM is just the residue of the active representation tokens. This means that the composition presented here is destructive. Because the space of hidden-layer units is typically very large, each compound should usually be distinct, but it is thoroughly possible that several different composite functions will closely overlap in some region in the hidden-layer space, in which case the exact constituents will be unrecoverable even in principle. This happens because the tokens are actions; once an action is taken it is gone. Since the composite representations are themselves actions (and therefore extended in time), their components need not be concurrently present.

Discussion

There are a number of different regularities which traffic under the systematicity label; compositionality has generally been invoked as an explanation for these regularities. I will very briefly mention a few such properties, and discuss implications for these arguments which result from a disposition-function based account of representation.

When a new word is heard for the first time, it often can be subsequently used appropriately not only in the context in which it appeared, but in many widely different contexts. Compositionality has often been invoked as an explanation for such one-shot learning. Whether or not SRNs may be made suitably systematic in their word acquisition (for hopeful signs that they may be, see Desai (2002a) or (2002b)), it is clear that SRNs are not always by their nature capable of an appropriate amount of systematic generalization; if they are formally compositional, then compositionality cannot bear the weight of explaining one-shot learning.

An even more basic application of systematicity has to do with novel arrangements of known words. A person who understands the sentence JLM will typically have little difficulty understanding, on first hearing, the sentence MLJ. This would be explained, the story goes, if the system utilized the very same representations in understanding the latter sentence that it had shown mastery of by comprehending the former. Compositionality of representation would explain an agent's abilities because the same (known) representations could be recombined in a novel order.

This story may be more complicated than it at first appears. The disposition functions which compose to form the network's representation of JLM will indeed be the very same functions which make up MLJ, but nevertheless, there is no guarantee that a network which responds appropriately to JLM will do so when given MLJ. There is a difference between appropriate representation and appropriate response: the latter constrains only the outputs in particular seen cases, while the former means processing in virtue of dispositionally correct intervening states. Since the network does not have an unambiguous way of dividing the representational burden of the compound phrase among its constituent elements, even a network which has learned JLM and represents it in virtue of its understanding of M, L, and J is likely to do so using non-robust representations of those constituents. Representations robust to novel orderings likely result from specialized learning or input schemes, but do not generally follow from nor obviously require compositional representations.

Another abberant property of this compositionality is that it is associative (in the mathematical sense) and word-based. There is no analogue here to the natural syntactic complexity of language, e.g, words grouping to form intermediate phrases and clauses, which are in their turn composed into sentences. Since the suggested composition operator is associative, every conjunction of words has a valid representation in the network. Jackendoff's (1983) conclusion that

"every major phrasal constituent in a sentence corresponds to a conceptual constituent in the semantic structure of the sentence" (p. 76) reflects an appealing intuition. The compositionality exhibited here is wholly unstructured, however.

By showing a notion of representation under which SRNs are classically compositional, I do not intend to imply that they are somehow mere implementations of classical systems. SRNs clearly differ from traditional symbolic architectures in interesting and complicated ways. Rather, I conclude that the interesting properties of SRNs do not result from having context-sensitive representations, but from some other factors, perhaps including active and non-discrete representations, or holistic statistical methods for learning and storing these representations, which allows the modification of one function to affect all other representations.

Since SRNs do not generally exhibit the various regularities known as systematicity, but appear to be classically compositional, compositionality alone clearly does not explain systematicity. Which of the following two conclusions to draw is perhaps a purely definitional issue: that compositionality is not formally well characterized by homomorphism, or that compositionality does not do much to explain systematicity. Regardless of how the terminology is carved, it is apparent that compositionality of representations (i.e., the property that representations of compound facts are compounds of the representations of those facts) does not provide the explanation of systematicities that it has usually been accorded. What I think this means is that these perplexing regularities require some more stringent implementational explanation than compositionality, such as a discreteness of the mechanisms underlying the symbols, as has been suggested by Davies (1991), or possibly a particular requirement on compositionality, such as demanding that it respect the apparent hierarchical structure of the target pattern. More analysis of particular networks is probably needed to determine the actual conditions which will guarantee the appropriate regularities that are usually attributed to compositionality.

References

Aydede, M. 1997. Language of thought: The connectionist contribution. *Minds and Machines* 7:57–101.

Davies, P. 1991. Concepts, connectionism, and the language of thought. In William Ramsey, Stephen Stich, D. R., ed., *Philosophy and Connectionist Theory*. Lawrence Erlbaum.

Desai, R. 2002a. Bootstrapping in miniature language acquisition. *Cognitive Systems Research* 3(1):15–23.

Desai, R. 2002b. Item-based language learning in children and connectionist networks. In 38th *Annual Conference of the Chicago Linguistic Society*.

Elman, J. 1990. Finding structure in time. *Cognitive Science* 14:179–211.

Elman, J. L. 1995. Language as a dynamical system. In Port, R. F., and Van Gelder, T., eds., *Mind as Motion:Explorations in the Dynamics of Cognition*. MIT Press. 195–223.

Fodor, J., and Lepore, E. 2002. Why meaning (probably) isn't conceptual role. In *The Compositionality Papers*. Oxford University Press.

Fodor, J. 1987. *Psychosemantics*. MIT Press.

Haugeland, J. 1985. *Artificial Intelligence: The Very Idea*. MIT Press.

Horwich, P. 1997. The composition of meanings. *The Philosophical Review* 106(4):503–532.

Jackendoff, R. 1983. *Semantics and Cognition*. MIT Press.

Markman, A. B., and Dietrich, E. 2000. Extending the classical view of representation. *Trends in Cognitive Science* 4(12):470–475.

Rodriguez, P.; Wiles, J.; and Elman, J. 1999. A recurrent neural network that learns to count. *Connection Science* 11(1):5–40.

Tabor, W., and Tanenhaus, M. K. 1999. Dynamical models of sentence processing. *Cognitive Science* 23(4):491–515.

Van Gelder, T. 1995. What might cognition be, if not computation? *Journal of Philosophy* 91:345–381.

Zadrozny, W. 1994. From compositional to systematic semantics. *Linguistics and Philosophy* 17:329–342.