Compositionality and Biologically Plausible Models

(Stewart, T. and C. Eliasmith. (in press) Compositionality and biologically plausible models. In W. Hinzen, E. Machery, and M. Werning (eds.) *Oxford Handbook of Compositionality*. Oxford University Press. Penultimate Draft)

Terrence Stewart and Chris Eliasmith

The breadth of this handbook demonstrates the diversity of approaches to compositionality that characterize current research. Understanding how structured representations are formed and manipulated has been a long-standing challenge for those investigating the complexities of cognition. Fodor and Pylyshyn (1988) and Jackendoff (2002) have provided detailed discussions of the problems faced by any theory purporting to describe how such systems can occur in the physical brain. In particular, neural cognitive theories must not only identify how to neurally instantiate the rapid construction and transformation of compositional structures, but also provide explanatory advantages over classical symbolic approaches.

Traditionally, cognitive theories have expressed their components using an artificial symbolic language, such as first-order predicate logic, e.g. `chased(dog, boy)`. The atoms in such representations are non-decomposable letter strings, e.g. **dog**, **chased**, and **boy**. Fodor and Pylyshyn (1988) call this a classical symbol system or classical cognitive architecture, and the defining characteristic is that the individual atoms explicitly appear in any overall structure in which they participate. This sense of 'classical architecture' is used throughout this article.

The technical problem of how symbolic representations, and the relations between such representations, can be accounted for in a neural approach has driven much of the discussion of neurally-inspired compositional models. The specific approaches discussed in this paper have all been shown to meet Jackendoff's challenges (de Kamps and van der Velde, 2006; Gayler 2003), which highlight potential difficulties for neural systems that are easily accounted for by a traditional classical system. However, each of these approaches continues to face two main criticisms: a) the architecture is uninteresting because it is an implementation of a classical system; b) the architecture is not biologically plausible. We take it as important to understand how the brain implements a classical system (if it indeed does), but agree that none of the past proposals are sufficiently biologically plausible. In this paper, we first discuss why implementation details may, in fact, be important for understanding cognitive behaviour. We then review the past approaches and present concerns about

biological plausibility. We conclude by presenting a new architecture that is not an implementation of a classical architecture, is able to explain the relevant behaviour, and is biologically plausible.

The Purpose of Neural Models

Fodor and McLaughlin (1990) suggest that if a neural theory merely demonstrates how to implement a classical symbol system using neurons, then this is actually an argument *against* the importance of the neural description. The fact that symbol systems are physically instantiated in neurons becomes a mere implementational detail, since there is a direct way to translate from the symbolic description to the more neurally plausible one. It might then be argued that, while the neural aspects of the theory identify *how* behaviour arises, they are not fundamentally important for understanding that behaviour. Classical symbol systems would continue to be seen as the right kinds of description for psychological processes.

However, it seems clear that there are explanatory advantages to having the neural level of description in addition to the purely classical one. A realistic neural explanation opens the door to a wealth of new methods for analyzing and investigating cognitive behaviour, such as fMRI, EEG, single-cell recordings, and the rest of modern neuroscience. Without such an explanation, there is no way to generate rigorous constraints from such evidence, and no way to create testable neurological predictions. Cutting ourselves off from this empirical data because of our theoretical commitments (e.g., that cognitive systems are classical) is a case of putting the cart before the horse. Indeed, a neural implementation of a classical system would strengthen the plausibility of the view enormously. However, this means that the biological realism of the proposed neural implementation is of the utmost importance: if the biological constraints are unrealistic then all that remains is a neurally implausible implementation of a classical architecture, of which there are many examples already.

It is also important to entertain the possibility that a neural cognitive theory might *not* be an implementation of a classical symbol system. Perhaps implementational details force us to reconsider our prior theoretical commitments. That is, while our best implementation may exhibit compositional behaviour in its ability to rapidly manipulate structures, it may not have certain properties that are fundamental to the classical symbol system approach: e.g., in classical symbol systems a structured representation contains, as constituents, explicit representations of each of its components (Fodor & Pylyshyn, 1988; Fodor and McLaughlin, 1990). Notably, some of the Vector Symbolic Architectures (VSAs) we discuss below do not meet this criteria, while still providing the compositional

characteristics required for explaining cognitive behaviour. Indeed, the neural theory we subsequently present (based on VSAs) not only provides interesting neurological constraints, it also relies on non-classical theoretical commitments.

Evaluating Neural Models

A cognitive model of compositionality can be analyzed at any of a number of levels, including the molecular, cellular, network, systems, behavioural, social, etc. levels. For the purposes of this discussion, however, we focus on two levels in particular (though we take the theories to be analyzable at many of these levels): the behavioural and the neural levels. We have chosen to narrow our focus in this way because these particular levels of analysis make clear the distinctions between available alternative theories.

Starting with behavioural constraints, we note that while compositionality is clearly a fundamental component of cognitive activity, it is equally clear that compositional behaviour is neither perfect nor unlimited. Complex nested ideas and long conjunctions of concepts are difficult or impossible for people to process all at once. The idea of *cognitive load* is extensively used in behavioral psychology to increase task difficulty until performance errors gradually increase to the desired level. Importantly, this tends to be a gradual effect; people do not perform perfectly well at one moment, only to fail completely in a slightly harder situation.

In the classical symbol system approach, this is considered to be an issue of 'competence' versus 'performance' (after Chomsky, 1965). That is, the underlying theory provides the capacity for arbitrarily complex compositions, but the limitations of the human cognitive system lead to less than perfect performance. This suggests that the best way to understand human compositional activity is to consider it to be an approximation of an ideal theoretical construct, much as modern computers are considered to be approximations of ideal Turing machines.

If a theory does not follow the classical approach, it may incorporate limits on compositionality at the theoretical level (i.e., regardless of implementational considerations). For example, some VSAs combine components in lossy, imperfect ways, while still maintaining the accuracy needed for structured, organized cognition, in many cases. This 'inaccuracy' is introduced at the theoretical level – it is a consequence of how representations and their processing are formally characterized – as opposed to being an implementational detail.

For both classical and non-classical approaches, similar constraints are provided by considerations of

neural implementation. Modern neuroscience has led to a wealth of knowledge about the details of neurons in various regions of the brain. We know that neurons are limited in term of their firing rate, exhibit a great deal of random variation in their firing, are generally highly promiscuous in terms of their connections, and are limited to about 100 billion in the human brain. Neural systems are also known to be highly robust, as neuron death occurs regularly without catastrophic consequences to the overall system.

Neural cognitive theories should conform to these constraints. This is especially true for classical theories, since the only advantage they have over non-neural theories involves comparison to measurements made on real physical neurons. The remainder of this article examines four different cognitive theories in terms of how well they conform to these known biological limitations, and how neural implementation leads to constraints on overall compositional behaviour. As will be seen, the classical symbolic approaches are problematic, while a non-classical Vector Symbolic Approach accounts for behavioral limitations via realistic neural constraints.

Classical Architectures

The three methods for implementing classical symbol systems in a neural architecture discussed here are all capable of meeting Jackendoff's criteria for compositionality. That is, they are able to represent symbols and relations between symbols using a connectionist approach. These models generally do this by explicitly representing each component within a structure, and then adding representations of the relations between these components.

LISA: Learning and Inference with Schemas and Analogies

Hummel and Holyoak have presented a series of papers describing their LISA model (Hummel, John E Holyoak,Keith J., 2003; J. E. Hummel, Burns, & Holyoak, 1994; J. E. Hummel & Holyoak, 1997). Their model is meant to account for various aspects of analogical reasoning, using a schema-based approach. This is common in classical symbol systems, and so their main contribution is showing how neurons can implement this classical architecture. The neural plausibility of the proposal is thus essential to its contribution to our understanding of cognition.

In LISA, a structured representation is constructed out of at least four levels of distributed and localist representations. The first level consists of localist *subsymbols* (e.g. **animal**, **furry**, **human**, etc.). The second level consists of localist units connected to a distributed network of subsymbols relevant to

defining the semantics of the second level *symbols* (e.g., **dog** is connected to **furry**, **animal**, etc.). The third level consists of localist *subproposition* nodes that bind roles to objects (e.g. **dog**+**chase-agent** to indicate that the dog is the chaser, not the entity being chased). The fourth and final level consists of localist proposition nodes that bind subpropositions to form whole *propositions* (e.g. **dog**+**chaser** combined with **cat**+**chased** results in `chase(dog,cat)`).

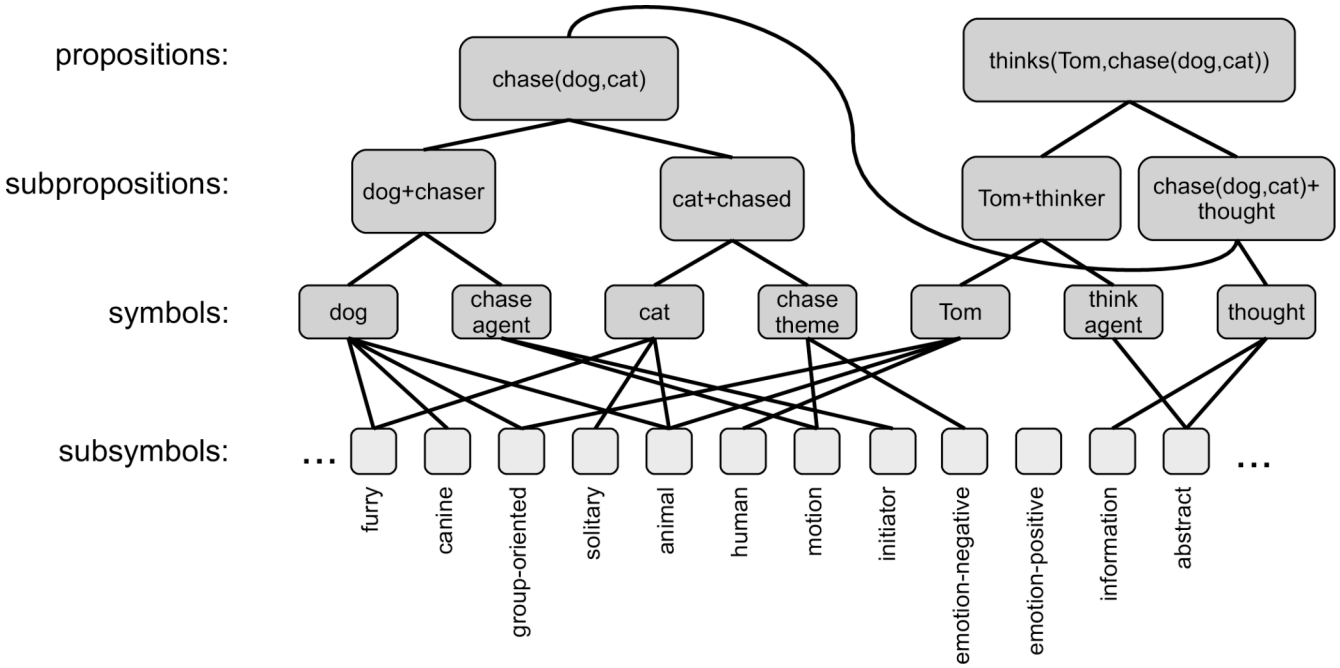

Figure 1: The LISA architecture. Each box is a single neural group. Shown are just those neural groups required to represent *dogs chase cats* and *Tom thinks that dogs chase cats*. Based on (Hummel & Holyoak, 2003, Figure 1)

Hummel and Holyoak (2003) are careful to note that each of their localist units is intended to be a population of neurons: "we assume that the localist units are realized neurally as small populations of neurons (as opposed to single neurons), in which the members of each population code very selectively for a single entity" (p. 223). Each such population only represents one subsymbol, symbol, subproposition, or proposition. They do not provide details as to how the neurons in this neural group interact to form this representation.

The simplest analysis in terms of neurological plausibility that can be done on this system is to determine how many neurons would be required to represent a reasonably complex language. If we

suppose that there are 4,000 nouns and 2,000 verbs (relations) in our language,[1] this suggests we need 6,000 populations to represent the basic concepts (i.e., at the second level of LISA's representational hierarchy). Assuming only two-place relations, we then need 4000*2000*2 = 16,000,000 populations to represent the third level (subpropositions), i.e. each noun playing agent or theme roles for each verb. At the fourth level (propositions) we need 2000*4000*4000 = 32,000,000,000 populations to be able to represent the possibility that, for each verb, any noun could be either an agent or a theme. Thus, to be able to represent any simple proposition of the form `relation(agent,theme)` requires around thirty billion neural groups, while only 100 billion neurons exist in the human brain. If higher order relations are desired as well (e.g. `knows(loves(agent, theme))`, `sees(hates(agent, theme))`, etc.), we need that same number of groups again for each such higher order relation.

Importantly, LISA does not fail gracefully when limited to a more reasonable number of neurons. If particular neural groups are not present in the architecture, then the corresponding structures *cannot be represented*. Even if some other mechanism were added that could identify neural groups that were not being used and adjust their connections to be able to represent the desired new concept, this would require adjusting synaptic connection weights between multiple neurons, a process which cannot occur in the few seconds it may take to read a novel sentence.

Another key aspect of the LISA model is its use of neural synchronization. At any given time, a few different propositions can be encoded in a LISA model. This is done by having the neurons corresponding to each proposition fire together, but at a different time from the other neurons. This idea is based on the currently heated debated among neuroscientists about the observations of such synchronized firing seen in physical brains. Since this bursting firing pattern is observed to have a period of around 25msec, and since neuron firing precision is considered to be around five milliseconds, this means that only five separate propositions can be encoded at the same time. Hummel and Holyoak take this to be a limit on human working memory.

However, the kind of synchronization used in LISA is not like that being argued for in biological brains. In LISA, synchronization occurs because there are inhibitory populations connected to each subproposition which set up an oscillatory behavior when the proposition they are connected to is given

---

[1] Estimates for the size of the vocabulary of English speakers vary between about 40,000-100,000. For instance, Crystal (2003) estimates that the average college graduate has 60,000 active words. While there are more nouns than verbs, there are about 11,000 verbs in English. As a result, the estimates we are using are very conservative.

a constant input. That oscillation is then reflected in all units that are excitatorily connected to these subpropositions (i.e. propositions and objects/relations). Usually, synchronization in the neurobiological literature is considered functional only if it is not explainable by common input. In LISA binding is established first by construction of subproposition units and that binding then results in synchronization. In the neurobiological literature, synchronization is supposed to *result* in binding (Engel, Fries, & Singer, 2001). Consequently, the neural plausibility of LISA is not supported by current work on synchronization. This severely challenges the claims to neural plausibility or realism made by the model's proponents. Our concern is that, if LISA adopted neurally plausible units, most of the explanatory mechanisms would fail to operate as they do in the much simplified cases explored to date.

LISA is able to represent complex nested structures, and it does so in a classical manner with populations of neurons that represent the sub-components of the overall structure. However, it is limited in terms of the depth of structures that can be represented. This limit is a hard, fixed limit that can be increased only by vastly increasing the number of neurons used. Indeed, even simple structures like `relation(agent,theme)` require more neurons than exist in the human brain. Furthermore, the neurons in this model are extreme idealizations and cannot be directly compared to real neurons. Although the synchronization aspect of LISA is inspired by neural evidence, it uses a mechanism that is at odds with the neurobiological literature. We believe these problems make LISA a poor candidate for neural explanations of cognitive behaviour.

Neural Blackboard Architectures

Many of the difficulties of the LISA model derive from the exponential growth in the number of neurons required. This problem is greatly reduced in van der Velde and de Kamps' (2006) neural blackboard architectures. This architecture consists of neural groups that can be temporarily bound to particular atomic concepts, and these neural groups can then be combined to form structures. Since structures are only built out of a restricted number of these temporary processors, this approach does not encounter the exponential growth problem of connecting every possible relation to every possible noun. By reuse of the structure assemblies, neural blackboard architectures can build much more complex structures than the fixed LISA approach with the same number of neural groups.

This approach uses a fixed number of *noun assemblies* and *verb assemblies* (plus separate assemblies for determiners, adjectives, prepositions, clauses, etc.). Any of these assemblies can be connected to

any of its associated words. That is, a particular noun assembly might at one time be bound to **boy**, while at another time it may be bound to **dog**. This binding is not done by forming new neural connections, as this would be implausible on a fast time scale. Instead, binding is accomplished via a complex mesh of carefully designed interacting neural groups (Figure 2b) that connect every noun assembly to every noun. This mesh requires eight neural groups for every noun/assembly pair (Figure 2c). That is, if there are 4000 nouns and we have 20 noun assemblies, 20*4000*8 = 640,000 neural groups are required. Similar calculations can be done for each of the other types of word assemblies.

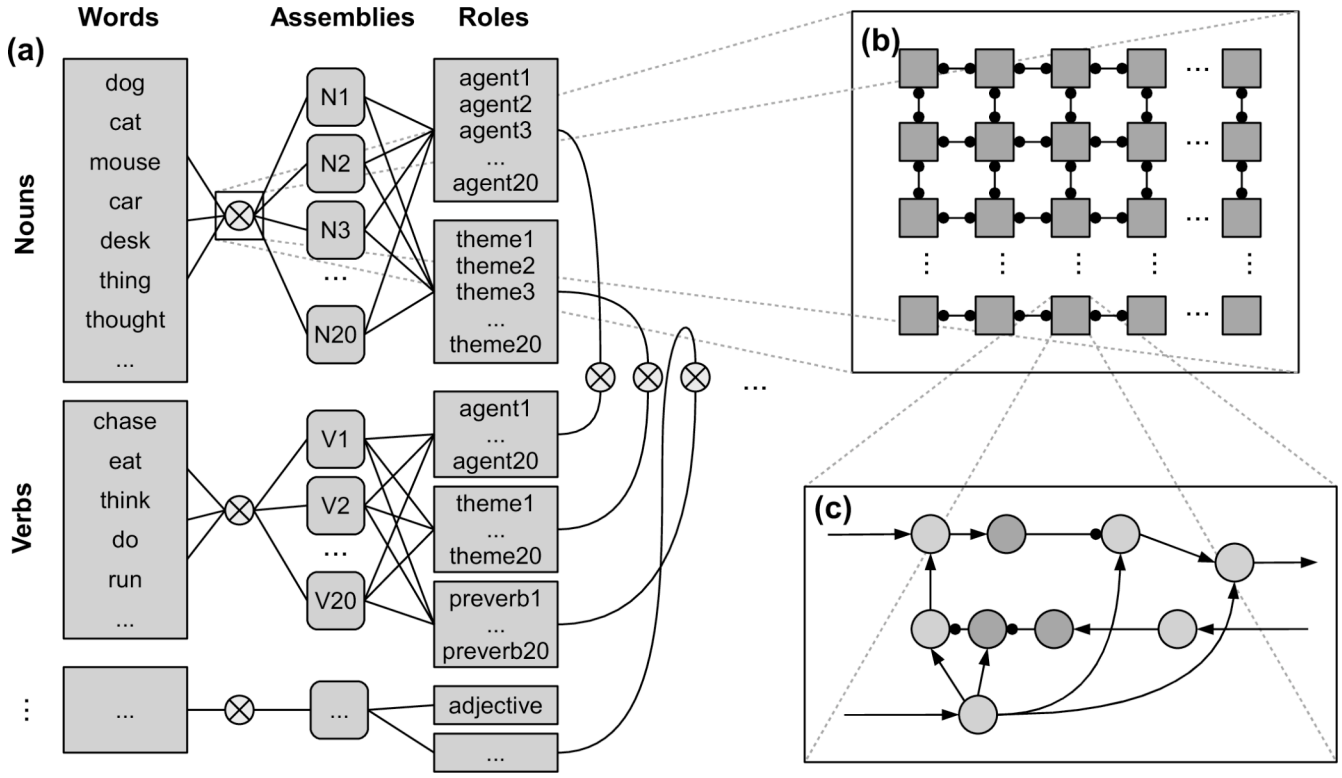

Figure 2: The neural blackboard architecture. Groups in (a) are connected by mesh grids (b) consisting of multiple copies of a complex system of neural groups (c). Excitatory connections are shown with arrows and inhibitory connections with circles. For more details, see (van der Velde & de Kamps, 2006).

Once these words are bound to particular assemblies in the blackboard, a separate set of neural structures is used to allow these atoms to bind together. This is done by having each of the assemblies also have connections to separate neural groups (called sub-assemblies) representing the *role* of the term. Thus each noun assembly has both an agent sub-assembly and a theme sub-assembly, which can be made active or non-active based on a control system that interacts with a gating assembly between

the two. Each of these sub-assemblies are connected to each other similar sub-assembly in the same manner as every noun assembly is connected to every noun. Given this system, it is possible to adjust the activations of the binding and gating systems to allow complex structures to be represented.

Unfortunately, this capability comes at a significant cost in terms of complexity. Notably, each gating circuit and memory circuit consists of eight or nine carefully arranged neural groups. The number of neurons needed for each group is not defined, but due to the degree of accuracy required in this complex structure, we estimate a minimum of about 100 neurons would be required per group, which would provide a signal to noise ratio of about 100:1, and a reasonably stable dynamics over about 2-5s (Eliasmith & Anderson, 2003). If we allow 20 assemblies for each word type and a total vocabulary (including nouns, verbs, adjectives, adverbs, etc.) of 50,000 words,[2] over 8,000,000 neural groups are required[3]. Given our estimate of at least 100 neurons per neural group, the architecture demands about in 800,000,000 neurons, or approximately 50cm^2 of cortex. While this is considerably less than is required by LISA, this is still a very large area of the brain (about the size of all language areas combined). And, this is the area that is required merely to *represent* a structure, it does not include the systems for controlling how sentences are encoded into this format, semantic connections between the words (encoded as subsymbols in LISA), methods for manipulating these structures, and so on.

Notably, the architecture depicted in Figure 2 has extremely dense inter-connectivity between neural groups. As far as we are aware, there is no evidence that such a dense connection arrangement is common across language areas of cortex. The evidence cited by van der Velde and de Kamps (2006) to support these structures demonstrates that some individual inhibitory cells in visual cortex synapse on other inhibitory cells in the same layer (Gonchar & Burkhalter, 1999) – it is not clear how this renders their architecture plausible in its details. After all, the blackboard architecture necessitates that all noun assemblies and all verb assemblies are connected, and that all nouns are connected to all noun assemblies. This demands very long distance, and highly complete connectivity, which is not observed in the brain. Most cortical connections are local and somewhat sparse (Song, Sjostrom, Reigl, Nelson,

---

[2] See ff 1 for references to typical vocabulary sizes in English.

[3] The neural requirements come from the two types of meshes shown in Figure 2. The first set of meshes binds words to assemblies and requires 50,000 * 20 connections, each of which requires 8 neural groups. The second set of internal meshes bind particular roles to each other to form the represented structure. For 20 different roles, this results in 20*20*20*8 neural groups. 50,000*20*8+20*20*20*8=8,064,000 neural groups.

& Chklovskii, 2005).

Furthermore, the highly complex binding and gating systems require thousands of intricately organized mutually dependent neural groups. This ensures that word assemblies are only bound to a single word at a time, and that various different possible structures involving the same words can be distinguished. If some of these neurons are removed, or if they are not connected in exactly the right manner, then it is unclear what behaviour will occur. Word assemblies could be stuck representing one particular word, or a particular noun assembly might become unable to connect to a particular verb assembly. This is not the sort of error generally associated with compositionality performance limitations.

The neural blackboard architecture also introduces a hard constraint on the number of noun, verb, and other word assemblies that exist. That is, if there are only 10 noun assemblies, then structures with exactly 10 nouns will be represented without difficulty, but a structure with 11 nouns will be impossible. This is not a pattern observed in human behaviour. To deal with this problem, NBA can increase the number of word assemblies to some sufficiently large number (50 or 100 has been suggested). This approximately linearly increases the number of neurons required (to 136cm^2 or 277cm^2). With this large a number of assemblies, it is possible that neural failure and timing issues may account for human performance limitations, although it is unclear to us how this occurs. However, this will require significantly more neural hardware than is associated with the language areas of the cortex.

The neural blackboard architecture is an improvement over LISA in that it has reduced the number of neurons required (though not necessarily to a plausible limit). However, the added complexity which allows this reduction does not seem to correspond to existing neural structures, and it is unclear what would happen to the system if neurons are removed or slightly mis-wired. In short, the system does not convincingly abide by neural-level constraints.

Tensor Products

Both LISA and the neural blackboard architecture follow a similar approach to representing classical structures: particular neural groups are set to represent the atoms, and neural connections of various forms represent how they are related. A radically different approach that employs *tensor products* was developed by Smolensky (1990). Considerable debate has arisen over whether this method is, in fact, equivalent to a classical symbol system. McLaughlin (1995) has convincingly argued that since the tensor product binding vectors (described below) can be chosen so that the representations of the

atomic constituents are present in the representation of the complete structure, tensor products should be considered to be implementations of classical symbol systems. It should be noted that Smolensky (1990) does not refer to his architecture as a classical system, but he seems to be employing a different, less common, definition than that we have adopted in this article. Our defition of a classical symbol systems as one which explicitly represents the constituents of a structure when representing that structure is consistent with the proposals of Fodor (1997), McLauglin (1995), and Jackendoff (2002).

The core idea behind the tensor product approach is to make use of a vector representation for the atomic components and to build up structures using algebraic manipulations. That is, instead of a particular neuron (or small neural group) representing **dog** and another neuron representing **cat** (as in LISA and neural blackboard architectures), a pattern of activity over many neurons forms the representation. For example, **dog** might be represented by the vector [0.4, 0.4, 0.5, 0.3, -0.4, …], while **cat** might be [0, -0.9, -0.2, 0.3, 0.2, …]. For technical reasons, these vectors are all fixed to have a magnitude of one, and thus lie on the unit hypersphere. We refer to the number of values in a vector as the *dimension* of that vector.

This vector representation can encode sub-symbols or semantic information about terms. In the simplest approach, the values in the vector might be various possible properties of the term, such as whether or not it is a living thing, whether it is furry, and so on. This is similar to the sub-symbols used in LISA, but any distributed representation scheme can be used. Importantly, the advantage offered by the tensor products approach is to define how such representations can be *combined* to form a structure. To create a structure representing `chase(dogs,cats)`, we perform the following calculation:

$$\text{dog} \otimes \text{agent} + \text{chase} \otimes \text{verb} + \text{cat} \otimes \text{theme}$$

To do this, the ⊗ operation is defined as the *tensor product* (i.e., outer product). This involves multiplying each element in the two vectors together to form a matrix of values, as shown in figure 3.

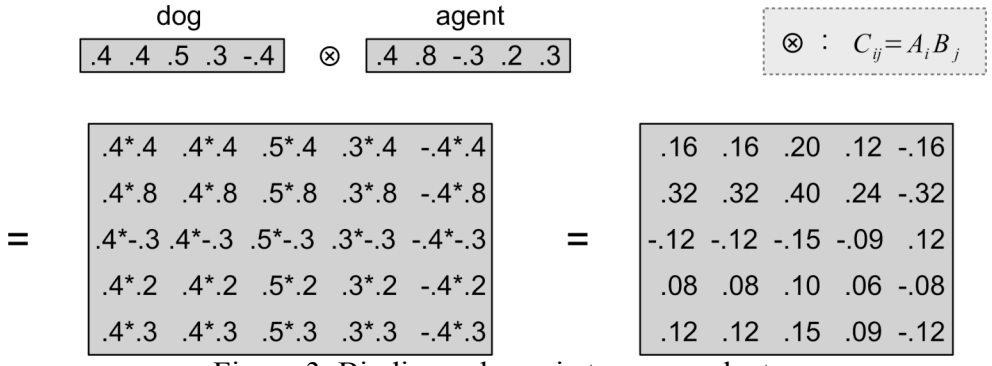

Figure 3: Binding values via tensor products

Importantly, with this technique the original components of the structure can later be extracted. That is, if we only have the overall representation matrix, we can determine what the original **agent** was by performing the *inner product* of the matrix with the value for **agent**. To complete our example, this can also be done for the **verb** and **theme** values, and the results summed to give the final matrix. In other words, this matrix is a representation of the entire structure, since the individual components can be recovered or decoded.

Although this approach is typically described in terms of vectors and algebraic manipulations, it can also be interpreted in terms of neurons. The values in the vectors or matrices can be encoded by the firing of a neural group, so a representation consists of a set of neural groups. The pattern of activation across the neural groups is the represented value. Encoding and decoding structure can be done by connecting groups together so that they calculate the outer or inner product. Since this requires multiplication of two values from different neural groups, there must be a neural mechanism capable of performing this nonlinear computation. There is some evidence that certain neurons can compute nonlinearities directly (e.g. Mel, 1994; Koch & Poggio, 1992), but it is currently a matter of considerable debate how common such mechanisms are in cortical neurons. However, it is also possible to use the Neural Engineering Framework (Elaismith & Anderson, 2003) discussed later in this article to organize highly typical cortical neurons to perform this multiplication.

The tensor product approach also takes into account a fundamental property of physical neurons: the fact that they are *noisy*. Because of the variability in spiking patterns and influences from the rest of the brain, it has been shown that the signal to noise ratio for a typical neuron is 10:1, meaning that it can only represent a value to within 10% accuracy (Rieke et al., 1997). If this constraint is taken into account when evaluating the tensor product approach, we find that it *degrades gracefully*. That is, it will slowly become less accurate, rather than suddenly failing like the architectures we previously considered. To demonstrate this, consider a case with 25,000 atomic terms in the language where we are representing a structure of the form `relation(agent,theme)`. We measure the accuracy of the representation by decoding the agent and determining which of the 25,000 atomic terms is closest to the resulting value. The accuracy shown in figure 4 indicates how often the correct decoding occurs. We can see from this figure that 20 dimensions (i.e. twenty values per vector) is sufficient to represent `relation(agent,theme)` with 95% accuracy, while 30 dimensions is required for more complex situations like `relation(A,B,C,D)` to be represented equally well.

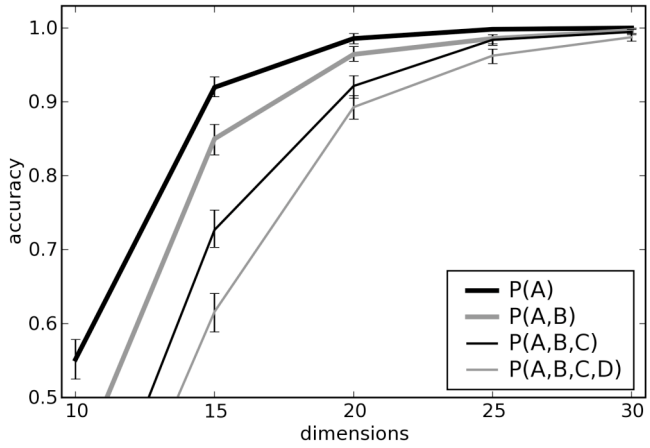

Figure 4: Decoding accuracy for relations of different complexities and number of dimensions, assuming representation noise of 10%.

This smooth degradation of accuracy leads to behavior where more errors are made the more complex a task is. This is a common pattern seen in behavioral psychology, where the cognitive load of a task is often increased specifically to cause errors in subject performance. In other words, tensor products fail in a manner similar to that of human behaviour, unlike the catastrophic failures seen for LISA and neural blackboard architectures.

However, tensor products encounter difficulties when creating more complex nested structures. In particular, for 20 dimensional vectors, at least 20 neurons are needed to represent a single atomic value, but 400 (20*20) are needed to represent the matrix for A⊗B, 8000 (20*20*20) are needed for A⊗B⊗C, and so on. In other words, the maximum depth of the structure is fixed by the number of neurons used, and this value grows exponentially. For two levels (e.g. 'The cat that the dog chased likes the mouse'), with 100 neurons per dimension (as for the blackboard architecture) and 3500 dimension per vector,[4] 1.2 billion neurons are required for each sentence. Clearly we will not be able to represent structures of sufficient complexity with the available neurons in the brain.

Tensor products are more realistic in terms of their behavioural limitations on compositionality, since attempting to build more and more complex structures leads to a gradual increase in error. Furthermore, when they are implemented using neurons, there is a natural way to model how the randomness of neural firing affects the high-level behavior of the system. However, tensor products

---

[4] See the section on Holographic Reduced Representations for the choice of this number of dimensions. Essentially, 3,500 neurons are needed to effectively code for 50,000 words.

require unrealistic numbers of neurons to represent deep structures, which makes them problematic as a neural theory of compositionality.

Non-Classical Architectures

The previous three approaches are all implementations of classical symbol systems. That is, their representations can be directly mapped to the standard symbolic approach to compositionality, where representations of the atomic components are constituents of the representation of the overall structure, just as **dog**, **chase**, and **cat** are constituents of `chase(dog,cat)`. The tensor product approach disguises this fact in its matrix representation, but it is possible to exactly extract those original components (ignoring implementation details such as neuron noise).

Recently, a number of new approaches have been developed which are similar to the tensor product approach, but which abandon the idea of being able to perfectly extract the original components. This family of approaches (including tensor products) are known as Vector Symbolic Architectures (VSAs; Gayler, 2003), and all share the basic principle of representing their atomic constituents via a numerical vector. They differ, however, in terms of what values are allowed, how vectors are combined together, and how the original values are extracted.

For the purposes of this article, we will focus on the VSA known as Holographic Reduced Representation (HRR; Plate, 2003). This makes use of atomic representation vectors of the same form as that used in the tensor product approach discussed above: vectors of numbers with a total length of one. Other VSAs, such as Binary Splatter Codes, only allow the values 0 and 1 for each dimension. Most of our discussion about how to form neural models will apply to any VSA. Plate (2003) provides a detailed overview of the algorithmic differences between numerous VSAs.

Holographic Reduced Representations

The key difference between HRRs and the tensor product approach is that in HRRs, *everything* is a vector with a fixed length. That is, instead of A⊗B producing a large matrix, it produces a vector of the same size as the original vectors. The operation used to do this is *cyclic convolution*, diagrammed in Figure 5.

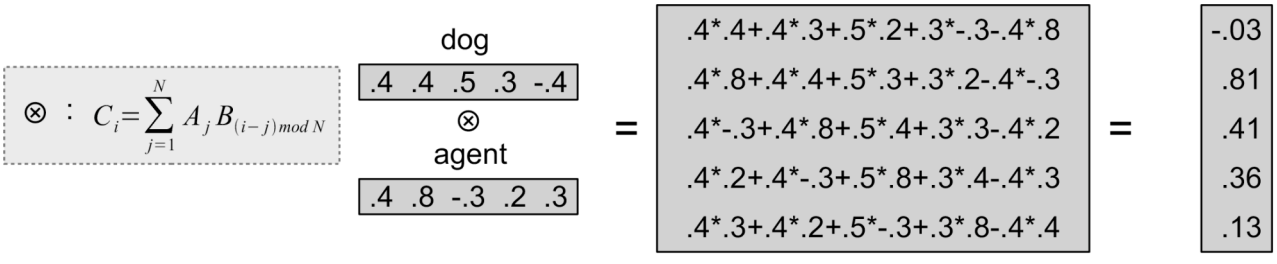

Figure 5: Binding values via Holographic Reduced Representations

Since the result is of the same dimension as the original vectors, we can make a representation as deep as is desired, while still requiring only a fixed number of neurons. However, this is accomplished at the expense of accurace: as the complexity of the structure increases, the expected accuracy of the decoding will decrease.

Decoding is accomplished by performing a cyclic convolution with the inverse of a value. The inverse is defined simply by rearranging the values in a vector so that, e.g., [a,b,c,d,e] becomes [a,e,d,c,b]. The result is a close approximation of the originally bound value. For example, if we have the representation

dog⊗agent+chase⊗verb+cat⊗theme,

we can perform the calculation shown in figure 6 to determine what the value that was originally bound to **agent**.

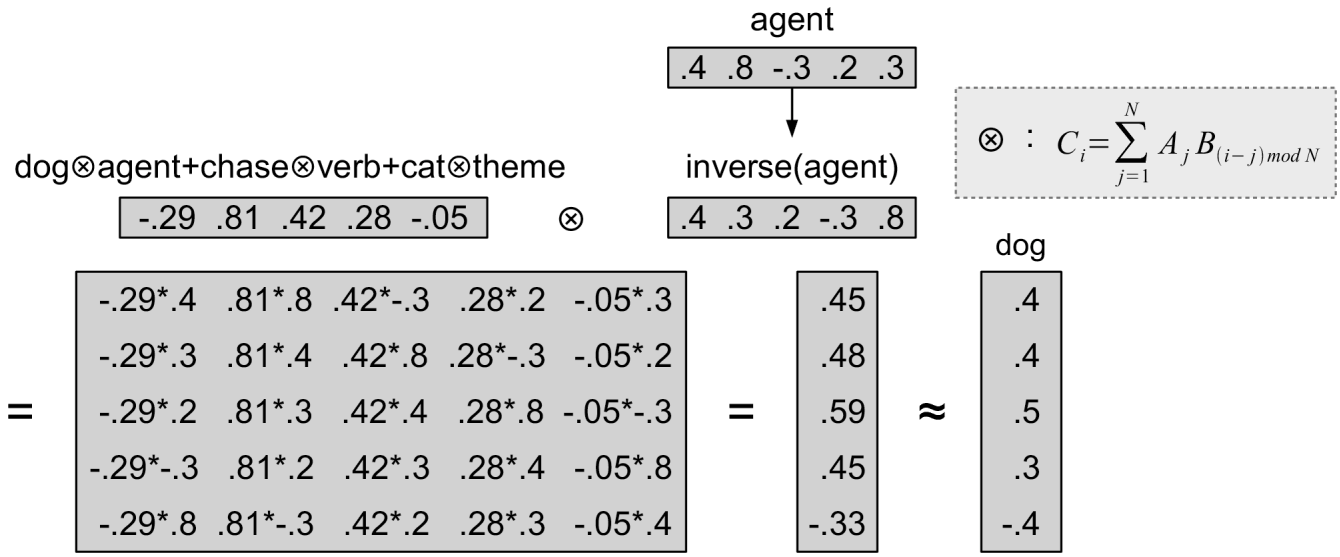

Figure 6: Extracting structure from an HRR representation. The combined structure is convolved with the inverse of **agent**, resulting in an output value that is an approximation of the original value for **dog**.

This works because of two fundamental properties of HRRs (and VSAs in general):

A⊗B⊗inverse(B)≈A and A+B ≈A.

In words, the convolution of a product of vectors with the inverse of one element is equal to the other element, and the superposition of two elements is somewhat similar to either element. As a

consequence, elements can be bound and superposed multiple times and still be recoverable from the resulting vector.

Plate (2003) determined how many dimensions are required to accurately represent and recover structures of this sort. For a fixed number of atomic values in the language (m) and a given maximum number of terms to be combined (k), and a certain probability of error (q), the following formula can be used to determine the number of dimensions needed (n).

$$n = 3.16(k - 0.25)\ln\left(\frac{m}{q^3}\right)$$

Given this, we can represent structures with up to 100 terms out of a vocabulary of 50,000 words with 99% accuracy using 3,500 dimensions. Since HRRs are non-classical representation systems, this limitation on accuracy will be a part of any theoretical discussion, even before considering the issues involved in implementing HRRs in neurons. However, using the estimate of 100 neurons per dimension, 350,000 realistic spiking neurons with properties typical to those found in the cortex would be sufficient. This requires less than $2mm^2$ of cortical area, significantly less than the neural blackboard architecture, LISA or tensor products. Indeed, this is many orders of magnitude fewer neurons than the other approaches. However, when implemented in noisy, realistic neurons, with complex structures and atomic vectors that are not randomly distributed, more dimensions are likely to be required to achieve this degree of accuracy. Exactly how much more depends on the neurophysiological details of the neurons involved. It should also be noted that current neural simulations have used 100 dimensions. Efforts are underway to scale this up.

Neural Engineering Framework

Given that HRRs have the best plausibilty of scaling appropriately, here we consider how to implement the necessary operations to construct HRRs in biological realistic networks. While these methods can be applied to the other approaches, their initial implausibility regarding the use of neural resources makes it unclear what value there is in pursuing that possibility.

Examing the implementation of HRRs in detail allows an analysis of how (or if) realistic neurons can perform the necessary calculations, and what effects different sorts of neurotransmitters, firing rates, and the other diverse features of real physical neurons might have on these representations. The approach we adopt is the Neural Engineering Framework (NEF; Eliasmith & Anderson, 2003), and has been used to model a wide variety of real neural systems, including the barn owl auditory system

(Fischer, 2005), the rodent navigation system (Conklin & Eliasmith, 2005), escape and swimming control in zebrafish (Kuo & Eliasmith, 2005), working memory systems (Singh & Eliasmith, 2006), and the translational vestibular ocular reflex in monkeys (Eliasmith, Westover, & Anderson, 2002).

As in the discussions of the previous models, neurons are divided into neural groups. However, in NEF, a neural group can represent a complete vector, rather than just one value within a vector. The neurons within a group are assumed to be *heterogeneous* (as observed in cortex) in that they all have different maximum firing rates and tuning curves, and possibly a variety of receptors and other physiological properties. The pattern of firing across these neurons can be characterized as a representation of a particular value, such as [0, -0.9, -0.2, 0.3, 0.2, …] (used above to represent the symbol cat). Notably, the number of dimensions in the vector is *not* the same as the number of neurons in the neural group. By adding more neurons, we increase the representation accuracy, counteracting the effects of random noise in neuron firing patterns.

To define a mapping from a particular value we want to represent to the population firing pattern, each neuron in the neural group is assigned an 'encoding vector,' which can be inferred from experimental data characterizing the tuning curve of a neuron if it is available. This encoding vector is a vector in the represented space for which the neuron will fire the strongest. This kind of characterization captures the observed behaviour of neurons in many areas of the brain, where such preferred direction vectors are generally found to cover all possible directions in the space being represented. For our purposes, we choose these to be random vectors for each neuron.

The details of how the encoding vector affects the firing of the neuron will vary depending on the type of neuron and the degree of accuracy to which the neuron is being simulated. In general, the activity a (i.e., the spike train) of a particular neuron i to represent a value $\boldsymbol{x}$ is:

$$a_i = G_i(\alpha_i \tilde{\phi}_i \cdot \boldsymbol{x} + J_i^{bias})$$

Here, α is the neuron gain or sensitivity, $\tilde{\phi}_i$ is the encoding vector, and J^{bias} is a fixed current to model background neural activity. G is the response function, which is determined by what sort of neuron is being modeled, including its particular resistances, capacitances, maximum firing rate, and so on. In our work, we use the response function for the leaky integrate-and-fire (LIF) model, which is widely used for its reasonable trade-off between realism and computational requirements. NEF can easily make use of more detailed models simply by changing this response function.

Using this approach, we can directly translate from a particular value that we want to represent ($\boldsymbol{x}$) to the steady-state firing rates of each neuron in the group (a_i). The value being represented is distributed across all of the neurons. This allows for any vector of a given length to be represented, and any value for each dimension within that vector, including negative numbers (which are problematic to encode just by naively considering the firing rates of neurons).

If we have the firing pattern for a neural group, we can also determine what value is currently being represented: the reverse of the encoding process. This is more complex than encoding, and in general it is impossible to perfectly recover the original value from the firing pattern. However, we can determine an *optimal linear decoder*, ϕ, to give a high quality estimate (Eliasmith & Anderson, 2003), $\hat{x}$:

$$\hat{x} = \sum_i \phi_i a_i \qquad \phi = \Gamma^{-1} Y \qquad \Gamma_{ij} = \iiint a_i a_j \, dx \qquad Y_j = \iiint a_j \boldsymbol{x} \, dx$$

This decoder can be constructed to be robust to random variations in the firing rates of neurons (and to neuron death). The representations can thus be made as accurate as desired by increasing the number of neurons used.

Since any theory of compositionality requires the ability to combine and extract information from the representational structures, we also need to determine how to manipulate these representations using neurons. This must be done via the synaptic connections between neural groups. We do not assume that it is possible to perform multiplication between two different values via a synaptic connection; instead, NEF shows how this can be performed using standard linear weighted connections.

Let us begin by considering the simplest case of a transformation that computes the identity function $f(x) = x$. That is, if we have two neural groups (A and B), and we set the value of group A to be $\boldsymbol{x}$, we want group B to also represent $\boldsymbol{x}$. This can be seen as the direct transmission of information from one location in the brain to another. For this situation, the optimal connection weights between each neuron i in group A and each neuron j in group B are (this can be seen by substituting the optimal decoding for neurons in A into the encoding equation for neurons in B):

$$\omega_{ji} = \alpha_j \tilde{\phi}_j \cdot \phi_i$$

If this formula is used to determine the strength of the synaptic connection between the neural groups, then group B will be driven to fire such that it represents the same value as group A. As noted in the previous section, the accuracy of this representation will be dependent on the number of neurons in the

groups. This system works even though none of the neurons in the two neural groups will have exactly the same encoding vector (and thus firing pattern). That is, there will generally not be a one-to-one correspondence between any neurons in the groups.

We can also connect neural groups in such a way as to transform the value from A to B. That is, we can set the synaptic weights so that B represents a vector that is, e.g., twice the vector in A, or the sum of the values in the A, or any other desired function *f(x)*. This is done using the same formula as above, but the decoding vectors ϕ are replaced by an *optimal linear function decoder* determined using the following formulae:

$$\hat{f(x)} = \sum_i \phi_i a_i \qquad \phi = \Gamma^{-1} Y \qquad \Gamma_{ij} = \iiint a_i a_j dx \qquad Y_j = \iiint a_j f(\boldsymbol{x}) dx$$

Using this approach, can determine the neural connection weights needed to compute the circular convolution of two input vectors. Thus, we can bind together the values in different neural groups to create any HRR structure. We have previously shown how this approach can be used to represent rule-following behaviour in different contexts by modeling the Wason card-flipping task (Eliasmith, 2005). This involved over 20,000 spiking neurons organized to perform cyclic convolution on values stored in an associative memory. In other words, not only can arbitrary structures be represented, but also manipulations of these structures can be represented and applied using this approach. This allows for fully compositional behaviour, as necessary to meet Jackendoff's (2002) challenges.

An important feature of the Neural Engineering Framework is that the methods for generating connection weights and representations continue to be applicable no matter how detailed the underlying models of single neurons is. It can be applied to rate neurons, leaky integrate-and-fire (LIF) neurons, adaptive LIF neurons, and even the highly complex compartmental models that require supercomputers to simulate the firing pattern of a single neuron. This means that as we obtain more information about particular neurons involved in a cognitive behaviour, we can add relevant information into the cognitive model and determine the effects of those insights on the overall model. Furthermore, simulations can first be done using a simplistic neural model requiring less computing power, and then once a suitable cognitive model is created a more detailed neural model can be used to generate precise predictions about firing patterns, representational accuracy, etc.

An example of adding increased biological detail involves the synaptic connection weights. In general, the approach described above results in both positively and negatively weighted connections. This is not consistent with what is sometimes called 'Dale's Principle,' the observation that in real brains

positive (excitatory) and negative (inhibitory) weights use different neurotransmitters and are attributable to distinct types of neurons. Parisien, Anderson, & Eliasmith (in press) show a related approach to determining weights which separates excitatory and inhibitory connections as needed, though with a slight increase in the number of neurons. This biological detail can be added to any NEF model, without disrupting the original function of the model.

Being able to incorporate whatever biological detail is deemed relevant for understanding the system allows the NEF to be a flexible tool for modeling neural systems. Coupling the NEF with the HRR approach leads to a neural model of compositionality that is consistent with available modern neuroscientific evidence as to the capabilities and limitations of real physical neurons.

Evaluating NEF HRRs

The result of implementing Holographic Reduced Representations using the Neural Engineering Framework is a detailed, biologically plausible model of compositionality. Unlike LISA and the neural blackboard architecture, an HRR-based system gradually becomes less accurate as the complexity of the structures increases. This matches the observed gradual increase in error as cognitive load increases. Like tensor products, similar behaviour is observed if neurons are destroyed in the NEF model. The NEF approach to representing values by encoding them in neuron firing patterns is highly robust both to increased noise and the loss of neurons. For example, in our model of the Wason card task (Eliasmith, 2005), on average a full third of the neurons could be removed from the HRR representation before the system became incapable of correctly decoding and applying structured rules. This is a side effect due to the system being designed to deal with realistic spiking neurons and neural variability.

The NEF provides a direct method for designing neural systems that can transmit HRR representations from one location in the brain to another, something not considered by the other approaches. Also, the algebraic manipulations of HRRs can all be implemented by calculating the connection weights between neural groups. For example, the operation $A \otimes B = C$ can be implemented by having a neural group representing A, a neural group representing B, an intermediate combined representation, and an final neural group representing C. The total number of neurons required is five times the number of neurons needed for a single representation. Given our previous calculations, this means about 1.4 million neurons would be needed ($9mm^2$ of cortex). However, this same population of neurons can be used for every binding and unbinding operation, so there is no need to scale the network as structures

become more complex, or as the number of possible elements increase. This allows the model to extract required parts of structures and to build up new structures as needed. All of these systems inherit the NEF's capacity for graceful degradation of performance as structure complexity increases.

Finally, since models created using the NEF can be made to be as realistic as possible (in terms of accurately modeling neural behaviour), the results of such models can be directly compared to available of neuroscientific evidence. Such comparisons could be based on patterns of connectivity, variability in firing rates, dendritic activity, and so on. This provides a potentially rich source of evidence for testing and comparing theories of compositionality.

However, this is not to say that our approach represents a full and complete theory of compositionality. Indeed, there are many unanswered questions that are topics of ongoing research. For one, there are questions about constructing appropriate vector representations corresponding to the underlying symbols in our system. We do not following the standard approach of having particular neural groups represent particular symbols (i.e. “grandmother cells”), but instead we claim that symbols correspond to distributed patterns of activation. However, this raises the question of how these particular patterns come into such correspondence, and how various parts of the brain maintain common representations. It should be noted that although we have assumed random patterns in this paper, we take these patterns to often include semantic similarity, so that the patterns for `cat` and `dog` would be similar in some important ways. Furthermore, our approach allows the dimensionality of the representation to change across different regions of the brain – certain regions need less accuracy or a less broad range of symbols. Exactly how this is accomplished is an open question.

The most important question, however, is how such a compositional system can be controlled. In this paper we have focused entirely on the question of representation, and ensuring that the representation would support compositional structure manipulations. To make use of such a system within a full cognitive architecture it is important to specify how this facility is used to answer questions, process complex embedded sentences, form new grounded representations, and so on. Although we have made some progress in this direction, including using this approach to implement a production system associated with the basal ganglia (Stewart & Eliasmith, 2008), more work needs to be done. That said, we believe that the approach of combining Vector Symbolic Architectures with the Neural Engineering Framework resolves many implementation issues and offers an alternative perspective from a purely classical approach.

Summary

We believe that neurobiological constraints can, and should, inform theory choice when evaluating theories of compositionality. In particular, we find that by examining how a particular theory would be implemented neurally we can identify whether a model is implausible in terms of neural requirements (i.e. too many neurons, implausible connectivity, etc.). We can also determine whether the high-level behavior of a model due to neural restrictions is comparable to the performance limitations of compositionality observed in humans.

Examining LISA and neural blackboard architectures suggests that a direct implementation of a classical symbol system will inevitably be unrealistic. LISA requires many orders of magnitude more neurons than are found in the human brain. The neural blackboard approach requires fewer neurons, but in a highly complex and intricate arrangement that is unlikely to be robust. More importantly, neither approach exhibits the gradual degradation of performance as complexity increases that is characteristic of human behaviour.

Moving away from the directly classical approaches, the tensor product approach and VSAs in general (including HRRs) provide exactly the graceful degradation that is desired. Tensor products (which are arguably isomorphic to classical symbol systems), however, require an unrealistic number of neurons to capture the necessary structures found in language. Holographic Reduced Representations provide the best of both worlds: realistic neural limitations and realistic performance limitations.

Although HRRs exhibit compositional behaviour, they are not classical symbol systems. Even if the implementational details are ignored, a theoretical investigation of HRRs will diverge from classical theories of compositionality. In particular, the representations of the constituents of a structure are not present in the representation of the structure itself. Instead, noisy versions of these constituents must be extracted via algebraic manipulating: i.e., extraction will always provide merely an approximation of the original constituents.

This new theory of compositionality also provides new avenues for evaluation. Numerical comparisons can be made between the accuracy of this system as complexity increases and the accuracy observed in people. The model can also be used to generate predictions of what sorts of firing patterns would be observed in neurons performing this sort of task, what connectivity they would have, and even the amount of time it would take to perform structure manipulations. We believe that this exploiting these sources of evidence will be fruitful for evaluating theories of human compositional

behaviour.

References

Chomsky (1965). *Aspects of the Theory of Syntax*. Cambridge: The MIT Press.

Conklin, J., & Eliasmith, C. (2005). An attractor network model of path integration in the rat. *Journal of computational neuroscience, 18*, 183-203.

Crystal, D. (2003) *Cambridge encyclopedia of the English language*. Cambridge University Press. Cambridge, UK

Eliasmith, C. (2005). Cognition with neurons: A large-scale, biologically realistic model of the Wason task. In G. Bara, L. Barsalou, and M. Bucciarelli (Eds)., *Proceedings of the 27 th Annual Meeting of the Cognitive Science Society.* Stresa , Italy

Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation and dynamics in neurobiological systems.* Cambridge, MA: MIT Press.

Eliasmith, C., Westover, M. B., & Anderson, C. H. (2002). A general framework for neurobiological modeling: An application to the vestibular system. *Neurocomputing, 46,* 1071-1076.

Engel, A. K., Fries, P., & Singer, W. (2001). Dynamic predictions: Oscillations and synchrony in top-down processing. *Nature reviews: Neuroscience, 2*(10), 704-716.

Fischer, B. (2005). *A model of the computations leading to a representation of auditory space in the midbrain of the barn owl.* PhD thesis. Washington University in St. Louis.

Fodor, J., & McLaughlin, B. (1990). Connectionism and the problem of systematicity: Why smolensky's solution doesn't work. *Cognition, 35,* 183-204.

Fodor, J., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28*, 3-71.

Gayler, R. (2003). Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. *ICCS/ASCS International Conference on Cognitive Science*, Sydney, Australia: University of New South Wales. 133-138.

Gonchar, Y. & Burkhalter, A. (1999) Connectivity of GABAergic calretinin-immunoreactive neurons in rat primary visual cortex. *Cerebral Cortex 9,* 683-696.

Hummel, J. E., Burns, B., & Holyoak, K. J. (1994). Analogical mapping by dynamic binding:

Preliminary investigations. In K. J. Holyoak, & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory: Analogical connections*. Norwood, NJ: Ablex.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review, 104*(3), 427-466.

Hummel, J. E., & Holyoak, K. J. (2003). A symbolic-connectionist theory of relational inference and generalization. *Psychological Review, 110*(2), 220-264.

Jackendoff, R. (2002). *Foundations of language: Brain, meaning, grammar, evolution.* Oxford University Press.

Koch, C. & T. Poggio (1992). Multiplying with synapses and neurons. In T. McKenna, J. Davis & S. F. Zornetzer (Eds.) *Single Neuron Computation*. Boston, MA, Academic Press.

Kuo, D., & Eliasmith, C. (2005). Integrating behavioral and neural data in a model of zebrafish network interaction. *Biological Cybernetics, 93*(3), 178-187.

McLaughlin, B. (1995) Classical Constituents in Smolensky's ICS Architecture. In M.L.D. Chiara, K. Doets, D. Mundici, & J. van Bentham (Eds.) *Structures and Norms in Science*. Kluwer Academic Publishers.

Mel, B. W. (1994). Information processing in dendritic trees. *Neural Computation, 6*(6), 1031-1085.

Parisien, C., C. H. Anderson, & C. Eliasmith (in press). Solving the problem of negative synaptic weights in cortical models. *Neural Computation.*

Plate, T. (2003). *Holographic reduced representations.* Stanford, CA: CSLI Publication.

Rieke, F., Warland, D., de Ruyter van Steveninck,R., & Bialek, W. (1997). *Spikes: Exploring the neural code.* Cambridge, MA: MIT Press.

Singh, R., & Eliasmith, C. (2006). Higher-dimensional neurons explain the tuning and dynamics of working memory cells. *Journal of Neuroscience, 26,* 3667-3678.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence, 46,* 159-217.

Song, S., Sjostrom, P. J., Reigl, M., Nelson, S., & Chklovskii, D. B. (2005). Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biology, 3*(3).

Stewart, T.C., & Eliasmith, C. (2008). Building Production Systems with Realistic Spiking Neurons. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Meeting of the*

Cognitive Science Society. 1759-1764. Austin, TX: Cognitive Science Society.

van der Velde, F., & de Kamps, M. (2006). Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences, 29*, 37-70.