Connections, Binding, Unification and Analogical Promiscuity

Ross W. Gayler **Roger Wales**
Department of Psychology, The University of Melbourne
Parkville VIC 3052, AUSTRALIA
r.gayler@psych.unimelb.edu.au r.wales@psych.unimelb.edu.au

Abstract

This paper claims that higher cognition implemented by a connectionist system will be essentially analogical, with analogical mapping by continuous systematic substitution as the core cognitive process. The centrality of analogy is argued to be necessary in order for a connectionist system to use representations that are effectively symbolic. In turn, these representations are argued to be a necessary consequence of a sequence of broad design decisions needed to address technical problems in adapting a connectionist system for higher cognition. The design decisions are driven by the demands of a paradigmatic cognitive task and the desire to remain faithful to the constraints of connectionist components. Thus, the argument explains the origin of symbolic representations and analogy as necessary consequences of task demands and connectionist processing capabilities.

Introduction

One of the more persistent problems in cognitive science is the reconciliation of the emergent functional properties of human cognition with the apparently much more limited functional capabilities of the neural systems that implement them. Higher cognition has been most successfully modelled in terms of symbolic computations that appear implausibly difficult to implement neurally. On the other hand, connectionist systems (the currently favoured paradigm for modelling the presumed computational processes of neural systems) appear to be neurally implementable but far less capable than symbolic systems of implementing the desired cognitive functions.

Despite the relative success of symbolic computation the shortcomings of classical Artificial Intelligence (based on symbolic computation) suggest that simply scaling up the size and speed of current symbolic systems will not yield the desired cognitive functions. One response to this situation is to focus on building connectionist systems. This action is based on taking human cognition as an existence proof for the possibility of implementing higher cognition with a connectionist system.

Some researchers have implemented classic symbolic architectures in connectionist systems. For example, Touretzky and Hinton (1988) built a Distributed Connectionist Production System. We have chosen not to follow this approach of implementing known symbolic processes because we believe it will be bound by the limitations of current symbolic models.

The problem of attempting to find a connectionist architecture with the desired cognitive properties can be cast as one of efficiently searching design space. Given the vast number of potential connectionist systems we need a strategy to guide our exploration of designs. We have chosen to be guided by the constraints imposed by connectionist computational elements and the problems to be solved by higher cognition. By remaining true to the connectionist raw material we hope to allow solutions that are obscured by taking symbolic operations as the primitive functions of processing. If it turns out that the emergent properties of such a connectionist system may be characterised as

---

We extend special thanks to Pentti Kanerva for providing stimulating discussions.

FINAL VERSION FOR PUBLICATION
ANALOGY'98, SOFIA

symbolic, then that is further evidence for the plausibility of the system (given the success of symbolic models - but not symbolic implementations).

In this paper our strategy for exploration of design space may be summarised by the question: Starting with a simple connectionist system; what minimal design decisions might lead to a capability for higher cognition?

This question arises from an evolutionary stance. It is taken as given that higher cognition is the function of an artefact built from neural components and designed through evolution to solve certain survival problems. Given the conservative nature of evolutionary design and that connectionism is an appropriate model of neural computation we believe that a sequence of minimal modifications starting from the simplest connectionist system will be sufficient to yield a system with the desired cognitive capabilities.

This paper suggests a series of design problems and broad design approaches to their solution. (We aspire to precise, implementable design choices, but that is work in progress.) Since the modifications to the connectionist architecture are constrained to be minimal, the hope is that the resultant architecture will be practically implementable. Furthermore, we argue that the symbolic properties of higher cognition and the centrality of analogy to cognition arise as necessary consequences of the design decisions motivated by connectionist problems. Thus, the argument (to the extent it is successful) explains the emergence of symbolic properties and the centrality of analogy.

The Design Problem

In order to specify the design problem that is the basis of this argument it is necessary to state the functional capabilities that are required of the final system and the design of the initial connectionist system.

Required Functional Capabilities

Specifying higher cognition in entirety is obviously too ambitious a sub-goal for this paper. The critical cognitive characteristics sought are embodied in the ability to follow novel instructions. To make this concrete we will settle on an arbitrary but paradigmatic task[1] to be carried out by the cognitive system. The task is to be able to follow novel instructions, such as:

> I will show you an artificially coloured picture of an animal and play the sound of an animal. If the sound belongs to the pictured animal you must name the colour of the pictured animal, otherwise name the animal that made the sound.

For the purposes of this paper the issues of language understanding required for comprehension of the instruction are ignored and we focus on complying with the instruction once it has been comprehended.

Initial Connectionist System

The system is to be implemented with typical connectionist units. That is, each unit may have multiple inputs and a single output. All inputs and outputs are to be graded, scalar quantities. The output is a nonlinear monotonic function of the weighted sum of the inputs or products of groups of the inputs.

The system is to have a fixed architecture. That is, the pattern of interconnection of units is not to vary or be constructed as a function of the current task. For example, this rules out the ACME model of analogical mapping (Holyoak & Thagard, 1989) as a permissible architecture because the neural net is constructed specifically for each problem.

The final constraint on the connectionist architecture arises from the nature of the task. The system is to implement the “top” level of cognition. Therefore, it must be capable of

[1] This paper was inspired by Hadley (1998). He used an example of following novel instructions to motivate his argument that most human mental skills must reside in separate connectionist modules and thereby instantiate a “classical architecture”.

integrating multiple sensory modalities. We assume that the full system will have other levels where the sensory modalities are processed separately. The boundary of the system of interest is to be expanded until it reaches the point at which the modalities are separately represented.

Representational Design Decisions

Representing Novel Concepts

The first design problem to address is representational flexibility. The paradigmatic task requires the creation of new concepts. At the very least it requires a concept of the instruction to be followed. Therefore, the system must be capable of representing arbitrary new concepts[2].

This constraint rules out local representations where each unit has a fixed meaning. If all the units are pre-allocated to concepts how can new concepts be represented? It also seems implausibly wasteful to have unallocated units waiting to be allocated to what may be ephemeral concepts.

This constraint can be avoided by using distributed representations (that is, the representation consists of the pattern of activities across the units). However, not all distributed representations avoid the problem. The same argument would apply to distributed representations where the individual units have fixed meanings as features. Any fixed allocation of meanings to units will limit the representational possibilities.

A related argument comes from the requirement that the system should integrate information from multiple modalities. Below the point of integration the information from separate modalities travels on separate pathways. Above the point of integration it would be possible to have disjoint segments of the representation devoted to different modalities, but this would be wasteful of representational resources (units).

It is possible to have information from separate modalities represented over the same units at different times provided that context information is available as part of the representation. This will allow the representation to be interpreted differently depending on the source.

This style of representation results in the units having context dependent meanings. The activities of individual units become meaningless unless they are able to be interpreted in the context of the activities in the other units in the representation[3].

Therefore, the first design decision is to use a distributed representation where the individual units do not have fixed meanings[4]. Kanerva (1995) has also argued that fixed feature representations are impractical for open-ended domains.

Immediate Learning

The next design problem arises from the need for immediate learning. The system must be able to learn[5] the novel concepts immediately from a single exposure. This rules out iterative weight adjustment techniques such as

[2] Any use of "concept" and "representation" begs many philosophical questions. For current purposes read "concept" as "a mental state" and "representation" as "a physical state standing for a mental state".

[3] Context dependency does not have to be all or none. At one end of the scale we can put representations where the unit activities can be interpreted in isolation. At the other end of the scale we have representations in which only the entire pattern of activations has significance. Between these extremes are representations where subsets of the activation pattern may be assigned meanings.

[4] The degree of context dependency involves trade-offs. At the context-independent end we restrict the representational capacity and flexibility. At the total pattern end any corruption of the pattern would completely change the meaning. We suspect that a good trade-off might exist not too far from the context-independent end of the scale where each unit is interpretable in the context of a small number of other units (relative to the total number of units) and participates in multiple, overlapping, meaningful subpatterns.

[5] By learning we mean changing the state of the system so that future occurrences of the novel concept are recognised. This changed state must be able to persist longer than the immediate span of attention.

backpropagation because they are too slow, typically requiring thousands of exposures[6].

What we want is that some specific output pattern (representing the novel concept) should be produced in response to a specific combination of input patterns. This is equivalent to saying that we want to associate the input and output patterns and to retrieve the output pattern given the inputs as a cue.

This can be achieved with binding operators for associating patterns and unbinding operators for retrieving components from bound patterns. Generically, if **b=bind(x,y)** then **unbind(b,x)=y** where **b**, **x**, and **y** are patterns. A variety of binding operators have been developed (Gayler, 1998; Kanerva, 1996; Plate, 1994; Smolensky, 1990)[7]. All the binding and unbinding operators are able to be implemented as connectionist primitives able to operate in a single time step.

In the example above the patterns **x** and **y** may be taken as input and output patterns respectively. The pattern **b** (which is able to be created in a single time step) represents the association of **x** and **y**. If **b** is present[8] in an environment where unbinding occurs automatically, the presentation of the input pattern **x** will result in the creation of the output pattern **y**.

The corresponding design decision is that the connectionist system should implement immediate learning as pattern association via bind and unbind operators.

[6] This is not to say that iterative weight adjustment procedures have no place in connectionist systems, only that they are inadequate for short term cognitive learning.
[7] Terminology differs between systems and there is some scope for confusion as each system has at least two distinct operators that might be called binding. We use the term 'binding' for what might best be called structural binding. In this case each of the components is structurally required (also called role/filler binding or attribute/value binding). We use the term 'bundling' for what might be called decorative binding. In this case each of the components is optional (for example, the slots of a frame). This is called superposition or chunking. (The latter term is used differently by different authors.)
[8] Binding addresses the issue of creating the association. Other connectionist mechanisms, able to operate in a single time step, exist to make the representation of the association persistent.

Compatibility of Representations

The next problem arises because the paradigmatic task requires close interaction of short term and long term knowledge. The task calls on pre-existing skills such as animal identification and requires their integration with the short term concepts of the task and the work in progress. Therefore, there is a requirement that the system is able to integrate short term and long term knowledge.

In a traditional connectionist system short term knowledge is usually implemented as activations of units and long term knowledge as connection weights. This implementation captures the relative persistence of the two types of knowledge. However, as abstract representations, the activation vector and weight matrix are incommensurable and can only indirectly influence each other via the processing function of the system. Our intuition is that the use of such different representations for short and long term knowledge will make integration difficult.

Therefore, the next design decision is to require short and long term knowledge to be represented (though not necessarily implemented) in the same way. That is, bindings in short and long term memory must have identical representations and have identical effects on operations.

These properties automatically come from binding methods that are based on element-wise multiplication operations of terms. In connectionist systems activations and weights interact multiplicatively. Therefore, in a binding operation short term knowledge (activations) and long term knowledge (weights) may be used interchangeably provided that they are all represented as vectors of the same dimension.

A newly created binding may be kept as an activation pattern or added into a weight vector with equivalent effect. Thus short and long term knowledge are identical in terms of their ability to be interrogated by current processing. The only asymmetry between the two storage forms is that whereas activation patterns may interact with other activation patterns and weight patterns, weight patterns may not interact with other weight patterns except via the mediation of an interaction with an activation pattern.

Discussion of Representations

Bindings as Virtual Neurons

Historically, single cell recording allowed neurophysiologists to identify stimuli that caused a neuron to fire. Over time researchers discovered cells responsive to progressively more complex stimuli. This was paralleled in the psychophysical literature by hypotheses of progressively more complex feature detectors. This led to the infamous "grandmother cell" as a *reductio ad absurdum* argument against complex feature detectors.

It seems implausible and wasteful that the brain might come prestocked with feature detectors for every concept that a person might possibly encounter (let alone what our descendants might encounter). This problem might be avoided if feature detectors could be created instantly on demand. This would be very difficult to achieve with real neurons but would be feasible if the feature detectors were virtual neurons implemented on a fixed neural base.

The design decisions so far are: that concepts will be represented by distributed patterns with individual units having no specific meaning; that association of concepts will be carried out by binding operators capable of immediate learning; and that short and long term representations of bindings will be equivalent, differing only in their form of storage and ability to interact with other bindings. These bindings may be thought of as virtual neurons. They are like neurons because for each binding an output pattern[9] may be retrieved by presentation of the associated input pattern. They are virtual because any number of neurons (bindings) may be implemented in a fixed number of real neurons or connectionist units (subject to soft capacity constraints).

The advantages of virtual neurons implemented as bindings are: that these neurons can be created on demand to represent novel concepts; that they can be created in a single exposure; that they may be ephemeral or permanent depending on whether they exist only as activation vectors or are stored in long term memory as weight changes.

Bindings and Systematicity

Fodor and Pylyshyn (1988) argued that one of the hallmarks of cognition that is explicable by a symbolic approach but not by connectionist models is systematicity. This is the property that having the capability to represent some concepts necessarily entails the capability to represent other related concepts. For example, they argued that a cognitive system able to think **Mary loves John** must necessarily be able to think **John loves Mary**.

Bindings automatically provide systematicity. If two patterns are bound together either may be used as a cue for the retrieval of the other. The notions of input and output, which are meaningful for real neurons, are not relevant to bindings as virtual neurons. In effect, whenever a binding implements a virtual neuron mapping **x** to **y** it also necessarily implements a virtual neuron mapping **y** to **x**.

It could be objected that producing all mappings between the bound patterns is not necessarily desirable. We agree, but discount this objection for two reasons. Firstly, we are concerned with higher cognitive functions. At this level we expect the major demand to be maximal exploitation of the available knowledge. Systematicity is a mechanism for generating hypotheses from prior knowledge to as yet unencountered situations. Halford

[9] Note that now the output is a pattern rather than a scalar value. This is a consequence of distributed representation rather than virtualisation. In a distributed representation the vector of scalar outputs of a group of units is the more convenient level of analysis of output.

(1996) has also argued for this capability which he labels “omni-directional access”.

The second reason is that we have assumed that where it is important to limit the hypotheses generated by systematicity this will be achieved by the details of the representational scheme. For example, Smolensky (1990) proposed a representational scheme in which fillers were bound to roles rather than directly to each other. For example, **John loves Mary** might be represented as **bundle(bind(lover,John), bind(loved,Mary))** where **lover** and **loved** are roles and **John** and **Mary** are fillers, rather than **bind(loves,John,Mary)** where **loves**, **John** and **Mary** are all fillers. In Smolensky’s representation the systematicity would be expressed with respect to role/filler pairs rather than directly between fillers.

Bindings as Rules on Constants

We have argued that bindings may be viewed as implementing virtual neurons. They may also be viewed as implementing rules or productions restricted to literal constants. A binding of **x** with **y** may be construed as implementing the rules IF **x** THEN **y** and IF **y** THEN **x**. Similarly for higher order bindings of **x**, **y**, and **z**: IF **bind(x,y)** THEN **z**, IF **bind(x,z)** THEN **y**, IF **x** THEN **bind(y,z)**, and so on. The restriction on the rules is that the antecedent and consequent consist only of literal constants because the bindings are between constant pattern vectors.

However, bindings do implement an extension relative to traditional symbolic rules. Because bindings are represented as pattern vectors they acquire some properties from vector arithmetic. The pattern vectors may be multiplicatively scaled and added. Thus it makes sense to talk about a binding operating on a mixture of patterns. In most binding systems **bind(x,y+z) = bind(x,y) + bind(x,z)**. It is also possible to have graded similarity between vectors. This allows rules implemented as bindings to operate in a graded fashion.

Processing Design Decisions

Generate Dissimilar Vectors

The next design decision is required as a consequence of an earlier decision. Recall that we decided to use distributed pattern vectors to represent new concepts. Every time a new concept is created a new pattern will be required to represent it. What constraints might exist on the patterns that can be used for new concepts?

Thinking of the binding as a virtual neuron there will be one or more input patterns to be associated with the output pattern representing the novel concept. The binding process imposes no constraint on the choice of output vector because any vectors may be bound[10]. The major constraint is that novel concepts require novel concept vectors. They should not be identical to any pre-existing concept vector otherwise the combination of inputs will be bound to a pre-existing concept.

In standard connectionist models much use is made of the fact that there is a graded similarity relation between vectors. To the extent that one vector is similar to another it is able to stand in for the other vector in further processing. If a new concept is truly novel the pattern representing it must not be similar to any pre-existing vectors in order to avoid having effects similar to some pre-existing concept.

Given immediate learning, pattern vectors for new concepts will be created at a point when the system is in a state of ignorance about the potential relationships between the new concept and any pre-existing concepts. Thus, even if the new concept vector should ideally be similar to some pre-existing concept vector it must necessarily be created dissimilar to all pre-existing concept vectors.

[10] There may be some exceptions. In multiplicative binding (Gayler, 1998) a pattern may not be bound to itself. This is not a problem in the current case because it implies that the output vector is identical to one of the input vectors (in which case the output vector has already been assigned to another concept).

The corresponding design decision is that vectors representing new concepts should be generated to have zero or minimal similarity to all pre-existing concept vectors. It will simply be assumed that such a generation mechanism is feasible.

Vector Generation Mechanisms

We discuss some possible generation mechanisms with no particular commitment to any of them. The least interesting possibility for creating dissimilar vectors is to rely on random noise in the system. Random high dimensional vectors have close to zero similarity on average. An extra mechanism might be required for those rare occasions when the new vector happened to be similar to an old vector.

Another mechanism (possibly an implementation of the previous one) is to modify a system with recurrent dynamics, similar to the Brain-State-in-a-Box (Anderson, Silverstein, Ritz, & Jones, 1977), such that pre-existing vectors become repellors in the state space. When presented with a novel stimulus the system would settle into a state different from any previously encountered state. The randomness in the choice of the new pattern might come from chaotic dynamics or the amplification of innate noise.

The most interesting possibility is that the novel concept vector might be created as a side effect of binding. For example, the pattern representing the binding (or some deterministic function of the binding pattern) of the inputs could be used as the output pattern. Multiplicative binding (Gayler, 1998) is essentially a randomising operation. The representation of **bind(x,y)** is not similar to either **x** or **y** when **x** and **y** are dissimilar (as we argue they should be if they represent concepts of higher cognition)[11]. Any novel combination of representations to be bound will necessarily generate a binding that is novel. This avoids the system having to decide when to create a new representation. If the combination is novel the binding and the output pattern will also be novel.

Vectors and Classical Symbols

The decision to represent novel concepts with vectors dissimilar to all pre-existing vectors leads to discrete representations. In general, two vector representations will either be identical or dissimilar[12].

The ability to bind arbitrary input and output representations means that the output representations can have arbitrary referents. The actual arbitrariness of the linkages will be guaranteed to the extent that representations for new concepts are generated at random.

Classical symbols are discrete and arbitrary, in that two symbols are either the same or different and that the form of a symbol has no necessary dependence on the referent of the symbol. Thus, the design decisions so far have led to connectionist representations that behave like classical symbols.

Systematic Vector Substitution

The design decisions taken so far have generated a new problem. If the representations of concepts are primarily dissimilar and arbitrary how can they be used? Traditional connectionist processing relies on the similarity of vector representations, which has been removed by the design decisions.

[11] In all the binding systems mentioned earlier (Gayler, 1998; Kanerva, 1996; Plate, 1994; Smolensky, 1990) the **bind()** operator reduces the similarity of compounds compared to their components. Considering the limiting case, the similarity of **bind(x,y)** to **x** and **y** is zero when the similarity of **x** and **y** is zero. Thus the overall effect of binding is to make representations less similar and the introduction of a single dissimilar pattern will render dissimilar every binding in which it participates. We are not denying the importance of those occasions when patterns and their bindings are similar, rather we are focusing on the occasions of dissimilarity because we believe they are more relevant to higher cognition and have been ignored by connectionists.

[12] It is possible for two representations to be similar and for processing to depend on that similarity. However, in a high dimensional vector space the proportion of vectors similar to any given vector becomes very small. Most pairs of vectors chosen at random will be dissimilar.

If the individual, isolated, pattern vectors do not carry information, what does? We believe that the information must be carried in the structural interrelationships of the bindings. During any episode the system will be creating many bindings which will be interrelated by the individual pattern vectors they have in common[13]. On a subsequent occasion the new episode will be recognised as equivalent to the previous episode if the structural interrelations of the bindings are the same (even though the pattern vectors composing the bindings may differ). This structural equivalence is proved if a systematic substitution of pattern vectors in the current episode yields the previous trace[14].

The corresponding design decision is that the basic mode of processing of the connectionist system should be continuous systematic substitution of representations for the elaboration of the currently active representations from previous representations.

Systematic substitution is relatively simple with the binding systems mentioned earlier. For example, in multiplicative binding, binding any structure to **bind(a,b)** will result in replacing all occurrences of **a** in the structure with **b** (and **b** with **a**)[15].

Rules with Variables

An interesting consequence of this decision is that it turns every constant pattern vector into a variable because every constant may be systematically substituted. Earlier it was noted that bindings could be thought of as rules restricted to constant terms. Without changing the representation systematic substitution allows every binding to function as a rule with variables. Thus, any encoding from any episode (even if encountered only once) becomes available as a generalised rule to the extent that it can be unified by systematic substitution with other structures.

It might be the case that in most circumstances systematic substitution is not needed because there is literal similarity between the representations. Processing based on literal similarity is equivalent to systematically substituting each pattern vector for itself. Thus, it could be the case that systematic substitution is continually occurring in the connectionist system but we can only detect it when we are operating in domains where literal similarity is not available.

Unification and Analogical Mapping

This design decision asserts that systematic substitution is a necessary component of the cognitive process because of the consequences of the earlier design decisions. Systematic substitution is at the heart of analogical mapping. Therefore, we are asserting that analogical mapping is a necessary component of the cognitive process.

It is worth expanding on the possibility that analogical mapping may be at the heart of cognition. We referred earlier to the problem of the system knowing when to create novel representations and suggested that one possibility is that the representations are functionally dependent on the inputs combined. That is, novel combinations of inputs would result in novel representations.

Given that very few situations are identical (especially if you consider the goals of the cognitive system as a representable component of the situation) this has the potential to make every representation a novel representation. The mechanism proposed here to overcome this is to use a continuous process of systematic substitution to unify[16]

[13] A representation **bundle(prop(loves,Chris,Pat), prop(loves,Pat,Robin), prop(loves,Robin,Chris))** would be structurally equivalent to any representation of the form **bundle(prop(l,c,p), prop(l,p,r), prop(l,r,c))**.

[14] Given the episode **bundle(prop(loves,Alex,,Jerry), prop(loves,,Jerry,Lee), prop(loves,Lee,Alex))** the systematic substitutions **{Alex⇨Chris, Jerry⇨Pat, Lee⇨Robin}** would transform it into the previous episode. Equivalently, the previous episode could be transformed to the current episode by systematic substitution. For current purposes, we ignore the representational details governing whether substitution for the relation (**loves**) is allowed and whether roles and fillers might be interchangeable.

[15] For other substitution mechanisms based on binding see Halford, Wilson, Guo, Gayler, Wiles & Stewart (1994), Kanerva (1997), and Plate (1997).

[16] Unification is a proof technique used in logic programming which uses substitution of variables to make terms equivalent. It is computationally expensive when

the current representation with all previously encountered representations.

We also referred earlier to the multiplexing of information from multiple modalities over the same representational resources. This imposed a requirement for context to be represented as a component of the content and made the interpretation of representations context dependent. As the number of contextual states increases this also has the effect of turning each representation into a novel representation (even if the entity being represented remains constant).

Therefore, it is possible that the cognitive system is thoroughly promiscuous in the generation of new representations and that these representations will appear to be random when viewed in isolation. How are these novel representations to be interpreted and acted upon if they appear random?

A similar problem arises in the technicalities of vector systems. Any vector can be decomposed into contributions from a set of basis vectors. By requiring new concept representations to be arbitrary and dissimilar, we have removed from the underlying hardware the possibility of having a distinguished basis set (fixed features) for decomposing composite structures (because the basis vectors are now the concept vectors which are not known until they are created).

When faced with a pattern vector, how does the system know whether it represents a new concept or some composite structure that may be decomposed into other representations? This is important because a composite structure needs to be exploited by integrating it with related knowledge, whereas a novel concept should not be spuriously integrated with prior knowledge.

The solution proposed is to decompose it with respect to the other structures that already exist in short and long term memory. Our intuition is that this might be carried out as a continuous process of activation spreading from the active representations in short term memory, through the inactive representations in long term memory, creating further activations in short term memory.

If the process of propagating activation simultaneously pursues many systematic substitutions a shower of new activations will be created. Those activations that are identical with or consistently extend the pre-existing activations will reinforce those patterns and themselves, while inconsistent mappings die out as noise or remain as suppressed alternative decompositions to pursue if the current one becomes inconsistent.

This process can be viewed as a competition between potential decompositions. The first and most consistent decomposition would be more successful than competing decompositions at creating the feedback activations to reinforce itself. Thus the decomposition that occurs is the (or a) correct interpretation of the structure (by virtue of its success). Other potential decompositions are incorrect interpretations of the structure because they were less successful at integrating the active representations and long term memory. This automatically achieves the desired result that a concept vector should be decomposed and integrated where ever possible.

Conclusion

We have suggested a series of connectionist design decisions that seem to follow naturally from the nature of the cognitive task by respecting the essence of connectionist computation. These decisions[17] are:

implemented by a symbolic algorithm. Weber (1992) developed a connectionist system (not based on the binding techniques discussed here) that implements unification in constant time.

[17] Connectionist systems exist demonstrating all but the last decision (which we believe is plausible within the current state of the art of systematic substitution by binding and unbinding). Even with an implementation of continuous systematic substitution all the design decisions will need to be integrated and there will be auxiliary problems to be solved before a connectionist model of higher cognition can be demonstrated.

- use distributed representations where the individual units do not have fixed meanings;
- implement immediate learning as pattern association via bind and unbind operators;
- bindings in short and long term memory must have identical representations;
- vectors representing new concepts should be dissimilar to all pre-existing concept vectors;
- the basic mode of processing should be continuous systematic substitution of pattern vectors.

As necessary consequences of these decisions the connectionist system will demonstrate behaviour typical of classical symbolic systems and place analogy as the primary cognitive process. Interpretation of these decisions suggests that cognition is promiscuously analogical and that the basic mechanism of cognition consists of a continuous process of unification through systematic substitution of currently active representations with each other, all representations in long term memory, and new representations being created. In effect, this is a continuous data mining operation on a massive scale.

References

Anderson, J. A., Silverstein, J. W., Ritz, S. A., & Jones, R. S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review*, *84*, 413-451

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

Gayler, R. W. (1998, July). *Multiplicative binding, representation operators, and analogy*. Poster session presented at Analogy'98, Sofia, Bulgaria.

Hadley, R. F. (1998). *Connectionism and novel combinations of skills: Implications for cognitive architecture* (Technical Report SFU CMPT TR 1998-01). Burnaby, Canada: Simon Fraser University, School of Computing Science.

Halford, G. S. (1996) Relational knowledge in higher cognitive processes. In *Relational knowledge in higher cognitive processes.* Symposium conducted at the XIVth Biennial Meeting of the International Society for the Study of Behavioral Development, Quebec City.

Halford, G. S., Wilson, W. H., Guo, J., Gayler, R. W., Wiles, J., & Stewart, J. E. M. (1994). Connectionist Implications for Processing Capacity Limitations in Analogies. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Analogical connections*. Norwood, NJ: Ablex.

Holyoak, K. J. & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, *13*, 295-355.

Kanerva, P. (1995). A family of binary spatter codes. In F. Fogelman-Soulié & P. Gallineri (Eds.), *ICANN '95, Proceedings International Conference on Artificial Neural Networks, Vol. 1* (pp. 517-522). Paris, France: EC2 & Cie.

Kanerva, P. (1996). Binary spatter-coding of ordered ***K***-tuples. In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, & B. Sendhoff (Eds.), *Artificial Neural Networks - ICANN 96* Proceedings 1996 International Conference, Bochum, Germany; *Lecture Notes in Computer Science* 1112), 869-873. Berlin: Springer.

Kanerva, P. (1997). Fully distributed representation. In *Proceedings RWC Symposium 1997 (Tokyo, Japan)*, 358-365.

Plate, T. A. (1994). *Distributed representations and nested compositional structure*. Ph.D. thesis, Department of Computer Science, University of Toronto.

Plate, T. A. (1997). Structure matching and transformation with distributed representations. In R. Sun & F.

Alexandre (Eds.), *Connectionist-Symbolic Integration*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*, 159-216.
Touretzky, D. S., & Hinton, G. E. (1988). A distributed connectionist production system. *Cognitive Science*, *12*, 423-466.
Weber, V. (1992). Connectionist unification with a distributed representation. In *International Joint Conference on Neural Networks IJCNN'92, Beijing, China*, 555-560.