

Realizing the Dual Route in a Single Route

Matt Goldrick, John Hale, Don Mathis, and Paul Smolensky
Department of Cognitive Science
Johns Hopkins University

poster presented at the 1999 Cognitive Science Society Meeting, Vancouver.

handout on

A Systematicity Bias for Associationist Learning

(1) The Problem of Systematicity

A. Systematicity of Mapping

- i. The default transformation in English past tense morphology involves "copying" much of the phonology of the present tense form into the past tense form. This operation is insensitive to previous exposure; even if a subject is given a nonword (e.g. [plemf]) containing a phoneme combination which has never been seen in a given position (e.g. [mf] in coda), they will still "copy" the phoneme into a novel inflected form ([plemft]).

This type of behavior has been called systematic: the response to a given phoneme is the same regardless of what position it occurs in. As it turns out, this is a very hard problem for simple associationist learning procedures.

- ii. A simple example (Marcus, 1998): The identity function.

Training examples:

Input	Output
1 0 1 0	1 0 1 0
0 0 0 0	0 0 0 0
1 0 0 0	1 0 0 0
1 1 1 0	1 1 1 0

Test item:

1 1 1 1	?
---------	---

- iii. Systematic output: 1 1 1 1

Generalization: the output position corresponding to input position i should have the same value as input position i .

- iv. Do simple associationist learning algorithms (e.g. the delta rule) enable connectionist networks to learn this generalization?

No.

Associationist Output: 1 1 1 0

- v. Why?

No training examples are ever seen in input position 4; therefore, no association can be established between position 4 and its corresponding output (Marcus, 1998).

- a. Delta rule: $\delta_{ij} = \alpha * (\text{target}_j - \text{output}_j) * \text{input}_i$
- b. Since no input is ever provided for input position 4, δ_{4j} will always be zero.

B. Distributed role vectors

- i. Critique: There was no reason the network should have generalized to position 4. Based strictly on the training data, there is no reason to assume that the systematic output is more correct than the associationist output. Assumptions that such generalization should occur are interpreter-imposed.

Suppose the first three units above represented the letters that made up some word, while the last unit represented the grammatical class of the same word. One would not assume that knowledge about mapping letters should transfer to the mapping of grammatical class information. We, the interpreters, are using more information in evaluating the network than we are providing it with.

- ii. What if the network was “told” that the untrained position was similar to the trained positions? The network might be able to take advantage of this similarity to generalize.

Following the PDP literature, we can represent similarity by using distributed representations. Does this move help us solve the problem of mapping systematicity?

- iii. Formally,
 - a. Assume tensor product input and output representations (Smolensky, 1990, 1995). These will consist of filler vectors (denoted by \mathbf{f}) bound to linearly independent role vectors (\mathbf{r}).

Def'n: input representation of symbol S in position 2:

$$\mathbf{f}^S \otimes \mathbf{r}^2$$

Def'n. Let \mathbf{T} be the tensor product of vectors \mathbf{f}^S and \mathbf{r}^2 . Let each element α, β of \mathbf{T} be equal to

$$\mathbf{f}_{\alpha}^S * \mathbf{r}_{\beta}^2.$$

For multiple roles and fillers, let

$$\mathbf{T}_{\alpha\beta} = \sum_Q \mathbf{f}_{\alpha}^Q \otimes \mathbf{r}_{\beta}^Q$$

for all roles Q .

—See Smolensky 1990, 1995 for an explication of the advantages of utilizing tensor product representations to provide structure to vectors.

b. Statement of the problem.

Given, as above, training data which says:

1. $\mathbf{f}^A \otimes \mathbf{r}^S \rightarrow \mathbf{f}^A \otimes \mathbf{r}^{f(S)}$ (filler A in position S maps to filler A in the corresponding output position $f(S)$).

2. $\mathbf{f}^B \otimes \mathbf{r}^Q \rightarrow \mathbf{f}^B \otimes \mathbf{r}^{f(Q)}$

Does the additional assumption that

3. $\mathbf{r}^S \cong \mathbf{r}^Q$ (roles S and Q are similar)

Allow associationist learning rules to generalize to unseen

$\mathbf{f}^A \otimes \mathbf{r}^Q \rightarrow \mathbf{f}^A \otimes \mathbf{r}^{f(Q)}$?

iv. Is there generalization at the limit of training?

a. Limit of delta rule training for two-layer network (Kohonen, 1977):

weight matrix $\mathbf{W} =$

$$\sum_j \mathbf{output}^J \cdot (\mathbf{v}^{J+})^T \quad (\sum_j \text{denotes sum over } j)$$

(\mathbf{v}^T denotes the transpose of \mathbf{v})

where \mathbf{v}^{J+} is the dual basis vector of input pattern J defined as:

$$\mathbf{v}^{I+} \cdot \mathbf{v}^J =$$

$$1 \text{ if } I = J$$

0 for all other J in the training set

b. \mathbf{W} after training on 1,2 above

$$\mathbf{W} = \mathbf{f}^A (\mathbf{f}^{A+})^T \otimes \mathbf{r}^{f(S)} (\mathbf{r}^{S+})^T + \mathbf{f}^B (\mathbf{f}^{B+})^T \otimes \mathbf{r}^{f(Q)} (\mathbf{r}^{Q+})^T$$

c. Test on unseen item

$$\mathbf{W} \cdot \mathbf{f}^A \otimes \mathbf{r}^Q \rightarrow 0$$

No generalization.

because

$$\begin{aligned} \mathbf{W} \cdot \mathbf{f}^A \otimes \mathbf{r}^Q &= \mathbf{f}^A (\mathbf{f}^{A+} \cdot \mathbf{f}^A) \otimes \mathbf{r}^{f(S)} (\mathbf{r}^{S+} \cdot \mathbf{r}^Q) && \text{(underlined terms are 0 by def'n of dual basis)} \\ &+ \mathbf{f}^B (\mathbf{f}^{B+} \cdot \mathbf{f}^A) \otimes \mathbf{r}^{f(Q)} (\mathbf{r}^{Q+} \cdot \mathbf{r}^Q) && \\ &= 0 && \\ &+ 0 && \end{aligned}$$

d. When the network has fully learned, the similarity of input roles does not matter; the learning algorithm will learn to completely dissociate one role from another role (and one filler from another filler).

v. Generalization before the limit of training?

a. Instead of training to the limit (which becomes harder as roles S and Q become more similar), we could stop training at some point where the similarity of roles still mattered.

b. Suppose we halt at a point where the dual basis vectors for the fillers have been learned, but not the dual basis for the roles.

Suppose that at this point in training:

$$W = \mathbf{f}^A (\mathbf{f}^{A+})^T \otimes \mathbf{r}^{f(S)} (\mathbf{r}^{S\oplus})^T \\ + \mathbf{f}^B (\mathbf{f}^{B+})^T \otimes \mathbf{r}^{f(Q)} (\mathbf{r}^{Q\oplus})^T$$

where

$$(\mathbf{r}^{S\oplus}) \cdot \mathbf{r}^S = 1$$

$$(\mathbf{r}^{S\oplus}) \cdot \mathbf{r}^Q = .75 \quad (\text{and the reverse for role } Q)$$

This is our representation of incomplete learning: the network is still sensitive to the input similarity of roles S and Q.

- c. Test on unseen item:

$$W \cdot \mathbf{f}^A \otimes \mathbf{r}^Q \rightarrow (.75) \mathbf{f}^A \otimes \mathbf{r}^{f(S)}: [.75 \text{ of filler } A \text{ bound to role } f(s)].$$

Generalization to wrong output role.

because

$$W \cdot \mathbf{f}^A \otimes \mathbf{r}^Q = \mathbf{f}^A (\mathbf{f}^{A+} \cdot \mathbf{f}^A) \otimes \mathbf{r}^{f(S)} (\mathbf{r}^{S+} \cdot \mathbf{r}^Q) \\ + \mathbf{f}^B (\mathbf{f}^{B+} \cdot \mathbf{f}^A) \otimes \mathbf{r}^{f(Q)} (\mathbf{r}^{Q+} \cdot \mathbf{r}^Q) \\ = \mathbf{f}^A (.75) \otimes \mathbf{r}^{f(S)}(1) \\ + \mathbf{f}^B (\mathbf{0}) \otimes \mathbf{r}^{f(Q)}(1) \\ = .75 \mathbf{f}^A \otimes \mathbf{r}^{f(S)}$$

Since the fillers have been completely learned, the second term is eliminated by the learning of the dual basis vectors.

- d. Assuming that $(\mathbf{f}^{B+}) \cdot \mathbf{f}^A$ is nonzero will not help us; this will produce a blend of

$$(.75) \mathbf{f}^A \otimes \mathbf{r}^{f(S)} \quad \text{and} \quad \mathbf{f}^B \otimes \mathbf{r}^{f(Q)}$$

In other words, a blend of the correct filler in the wrong output role and the incorrect filler in the right output role.

- e. When generalizing a transformation using associationist learning, being responsive to the similarity to a trained item's input structure necessarily entails being sensitive to the trained item's output structure.
- vi. This type of generalization is not what we desire. Distributed role vectors do not help solve the problem of systematicity of mapping.
—Need to incorporate structure sensitive biases into the learning algorithm.

(2) A Structure-Sensitive Search Heuristic for the Delta Rule

A. Structuring weights

- i. In order to impose structural constraints on weights, one must be able to recover structure from the weight matrix.
- ii. Assuming the weight matrix operates over tensor product representations, we can make use of the methods which are used to extract structure from tensor product representations.

- a. Here, assume that our tensor product representations only bind a single filler for each role. The general tensor product representation will then be a sum over all roles Q of the filler bound to role Q tensored with role Q .

$$\sum_Q \mathbf{f}^Q \otimes \mathbf{r}^Q$$

- b. If one wishes to extract the filler from role S , one needs to solve the following equation for the unknown binding vector \mathbf{v} :

$$\mathbf{v} \cdot \sum_Q \mathbf{f}^Q \otimes \mathbf{r}^Q = \mathbf{f}^S$$

The answer to this is $\mathbf{v} = \mathbf{r}^{S+}$. This is true because

$$\mathbf{r}^{S+} \cdot \mathbf{r}^Q = 0 \text{ for all } Q \neq S$$

$$\mathbf{r}^{S+} \cdot \mathbf{r}^S = 1 \text{ for } Q = S.$$

Since $\mathbf{r}^{S+} \cdot \mathbf{r}^S = 1$, the roles cancel out leaving just \mathbf{f}^S .

- c. We can extend this to the weight matrix. In order to implement a systematicity of mapping bias, we will need to know how fillers of particular input roles are mapped to fillers of particular output roles.

In the general case, the roles are distributed, and so the value of each input filler may be spread over a number of input units. The effect of each value of the input filler is therefore distributed over a number of weights. Similarly, the value of each output filler may be responsive to the value of a number of output units, each connected to a number of weights. We need to recover a non-distributed matrix $M^{f(S)S}$ specifying the mapping from fillers of S to fillers of $f(S)$. In other words, we wish to find the portion of the weight matrix W that performs a particular filler to filler mapping.

To do this, we take the dot product of the weight matrix with (1) the vector of input role S and (2) the dual basis vector of the output role $f(S)$.

$$\mathbf{r}^{f(S)+} \cdot \mathbf{r}^S \cdot W = M^{f(S)S}$$

The proof of this equation follows:

The response of the weight matrix to a particular filler bound to a particular role S^1 can be written as:

¹ To get the entire output of the network, this equation would have to be summed over all input roles S .

$$W \cdot \mathbf{f}^S \otimes \mathbf{r}^S = \sum_Q \mathbf{f}^{QS} \otimes \mathbf{r}^Q$$

where \mathbf{f}^{QS} is the correct output filler (as a result of input $\mathbf{f}^S \otimes \mathbf{r}^S$) bound to role Q .

By definition, $M^{QS} \cdot \mathbf{f}^S = \mathbf{f}^{QS}$; M performs the filler to filler mapping. So we may substitute this unknown into the equation.

$$W \cdot \mathbf{f}^S \otimes \mathbf{r}^S = \sum_Q M^{QS} \cdot \mathbf{f}^S \otimes \mathbf{r}^Q$$

We then multiply each side of the equation by $\mathbf{r}^{f(S)+}$. On the right hand side, this causes each term except for $Q=f(S)$ to be equal to zero (following the definition of the dual basis).

$$\mathbf{r}^{f(S)+} \cdot W \cdot \mathbf{f}^S \otimes \mathbf{r}^S = M^{f(S)S} \cdot \mathbf{f}^S$$

Since we are looking for a general solution for all fillers, we remove the \mathbf{f}^S term and reorder to yield the formula above.

$$\mathbf{r}^{f(S)+} \cdot \mathbf{r}^S \cdot W = M^{f(S)S}$$

This accomplishes the first part of transferring knowledge from one input–output mapping to another; first, we must know the knowledge that we wish to transfer. This knowledge is encapsulated in M .

- d. To finish the transfer, we must take the knowledge of the mapping and allow it to be applied to another input–output pair. This is done by use of the tensor product operation. Suppose we wish to rebind the matrix M to another input role Q and output role $f(Q)$. To do this, the unbound matrix M is tensored with the output role vector for $f(Q)$ and the dual basis vector for input role Q . This produces the transfer matrix

$$Z_{f(S)S}^{f(Q)Q} \text{ transferring the mapping } S \rightarrow f(S) \text{ to } Q \rightarrow f(Q).$$

$$M^{f(S)S} \otimes \mathbf{r}^{f(Q)} \otimes \mathbf{r}^{Q+} = Z_{f(S)S}^{f(Q)Q}$$

This can be shown by solving the extraction equation for the weight matrix term. Replace W with an unknown Z . We eliminate the $\mathbf{r}^{f(S)+}$ term by multiplying both sides of the equation by $\mathbf{r}^{f(S)}$. Similarly, we eliminate the \mathbf{r}^S by multiplying both sides by \mathbf{r}^{S+} , yielding the binding equation above.

- iv. Tensor products and the appropriate matrix operations allow us to isolate a specific portions of the weight matrix that determines the mapping for a particular role (or set of roles). We can also do the reverse, and take this role to role mapping and place it into other roles.

→ It should be noted that these operations generalize to non-local role vectors; as long as the roles are linearly independent, all of these operations will hold.

B. Making use of our structural operations

- i. Armed with knowledge of tensor products, we can now return to our original formulation of the systematicity of mapping problem (from (1) B iii. b.).

Given:

$$1. \mathbf{f}^A \otimes \mathbf{r}^S \rightarrow \mathbf{f}^A \otimes \mathbf{r}^{f(S)}$$

$$2. \mathbf{f}^B \otimes \mathbf{r}^Q \rightarrow \mathbf{f}^B \otimes \mathbf{r}^{f(Q)}$$

How can we generalize to unseen

$$\mathbf{f}^A \otimes \mathbf{r}^Q \rightarrow \mathbf{f}^A \otimes \mathbf{r}^{f(Q)} ?$$

- ii. Recall that δ -rule learning will provide the following matrix

$$\begin{aligned} \mathbf{W} = & \mathbf{f}^A (\mathbf{f}^{A+})^T \otimes \mathbf{r}^{f(S)} (\mathbf{r}^{S+})^T \\ & + \mathbf{f}^B (\mathbf{f}^{B+})^T \otimes \mathbf{r}^{f(Q)} (\mathbf{r}^{Q+})^T \end{aligned}$$

- iii. We need to extract the filler to filler mapping of input role S in order to transfer it to input role Q. We know how to map filler A to the correct output in role S, so we must transfer this knowledge to role Q. We do this following the methods above.

$$\mathbf{r}^{f(S)+} \cdot \mathbf{r}^S \cdot \mathbf{W} = \mathbf{M}^{f(S)S}$$

because

$$\begin{aligned} \mathbf{r}^{f(S)+} \cdot \mathbf{r}^S \cdot \mathbf{W} = & \mathbf{f}^A (\mathbf{f}^{A+})^T \otimes (\mathbf{r}^{f(S)+} \cdot \mathbf{r}^{f(S)}) (\mathbf{r}^S \cdot (\mathbf{r}^{S+})) \\ & + \mathbf{f}^B (\mathbf{f}^{B+})^T \otimes (\mathbf{r}^{f(S)+} \cdot \mathbf{r}^{f(Q)}) (\mathbf{r}^S \cdot (\mathbf{r}^{Q+})) \end{aligned}$$

$$= \mathbf{f}^A (\mathbf{f}^{A+})^T \otimes (1) (1)$$

$$+ \mathbf{f}^B (\mathbf{f}^{B+})^T \otimes (\mathbf{0}) (\mathbf{0})$$

$$= \mathbf{f}^A (\mathbf{f}^{A+})^T$$

$$\begin{aligned} \mathbf{M}^{f(S)S} \otimes \mathbf{r}^{f(Q)} \otimes \mathbf{r}^{Q+} = & \mathbf{Z}_{f(S)S}^{f(Q)Q} \\ = & \mathbf{f}^A (\mathbf{f}^{A+})^T \otimes \mathbf{r}^{f(Q)} (\mathbf{r}^{Q+})^T \end{aligned}$$

- iv. We can now add \mathbf{Z} to \mathbf{W} yielding \mathbf{W}'

$$\begin{aligned}
W' &= \mathbf{f}^A (\mathbf{f}^{A+})^T \otimes \mathbf{r}^{f(S)} (\mathbf{r}^{S+})^T \\
&+ \mathbf{f}^B (\mathbf{f}^{B+})^T \otimes \mathbf{r}^{f(Q)} (\mathbf{r}^{Q+})^T \\
&+ \mathbf{f}^A (\mathbf{f}^{A+})^T \otimes \mathbf{r}^{f(Q)} (\mathbf{r}^{Q+})^T
\end{aligned}$$

v. Test on unseen item:

$$W' \cdot \mathbf{f}^A \otimes \mathbf{r}^Q \rightarrow \mathbf{f}^A \otimes \mathbf{r}^{f(Q)}$$

Correct generalization.

because

$$\begin{aligned}
W' \cdot \mathbf{f}^A \otimes \mathbf{r}^Q &= \mathbf{f}^A (\mathbf{f}^{A+} \cdot \mathbf{f}^A) \otimes \mathbf{r}^{f(S)} (\mathbf{r}^{S+} \cdot \mathbf{r}^Q) \\
&+ \mathbf{f}^B (\mathbf{f}^{B+} \cdot \mathbf{f}^A) \otimes \mathbf{r}^{f(Q)} (\mathbf{r}^{Q+} \cdot \mathbf{r}^Q) \\
&+ \mathbf{f}^A (\mathbf{f}^{A+} \cdot \mathbf{f}^A) \otimes \mathbf{r}^{f(Q)} (\mathbf{r}^{Q+} \cdot \mathbf{r}^Q) \\
&= \mathbf{f}^A (\mathbf{1}) \otimes \mathbf{r}^{f(S)} (\mathbf{0}) \\
&+ \mathbf{f}^B (\mathbf{0}) \otimes \mathbf{r}^{f(Q)} (\mathbf{1}) \\
&+ \mathbf{f}^A (\mathbf{1}) \otimes \mathbf{r}^{f(Q)} (\mathbf{1}) \\
&= \mathbf{f}^A \otimes \mathbf{r}^{f(Q)}
\end{aligned}$$

vi. Appropriate use of the matrix operations above allows us to transfer an input role–output role mapping to different input and output roles. We have shown that this provides a mechanism for the transfer of systematicity of mapping.

C. Towards a learning algorithm making use of the transfer function

- i. We can include a bias in our learning algorithm that allows us to search for systematic solutions while minimizing error. Here we give a general picture of how such a search algorithm would work.
- ii. Let λ_{jl}^{ik} be the amount of transfer from the mapping of input role j to output role i to the mapping from input role l to output role k .

Transfer from input–output pair j,i to l,k occurs by extracting the matrices M^{ij} and M^{kl} . This provides us with the current mapping from j to i and l to k respectively. We can then attempt to make the mapping from l to k more similar to the j to i mapping by calculating

$$\Delta_{jl}^{ik} = M^{ij} - M^{kl}$$

This Δ will provide us with a direction in which to move the weights from l to k so that they are more similar to those from j to i . In the limit, the weights

from l to k will be identical to those from j to i. The mapping would then be exactly the same, making the two role completely systematic.

λ will then scale the Δ matrix. Once this scaling has been done, the Δ matrix will be appropriately re-bound into a full rank weight matrix and added onto the weights. This will alter only the targeted l to k weights.

There is an additional set of λ s which biases certain mappings towards zero.

These terms will be labeled λ_{Ol}^{Ok} . For these λ s, the target matrix will be a zero matrix. When these "zero λ s" are large, they will enforce a constraint that input roles map to only a few output roles (a kind of targeted weight decay for the superfluous connections).

- iii. We need a procedure for determining the values of the λ s. Intuitively, we can see that the value of the zero λ s is inversely related to the value of the other λ s.

For example, in the case of the identity map, input role j corresponds to output role j and to no other output role. In this case, λ_{Ol}^{Ok} should be large for all $k \neq l$. One can also see that in this case λ_{jl}^{ik} should be large for all cases where $j = i$ and $k = l$ (precisely the cases where λ_{Ol}^{Ok} is small).

We encode this intuition by the following procedure.

Let P be some function that assigns, for each input role j, a probability distribution over the mapping from j to each output role i. The probability of some mapping j to i will be the likelihood that the i is the "output correspondent" of j; i.e. $f(j)$. If we can assume a 1-1 function, then P might assign a probability of 1 to the most likely mapping for an input role and a probability of 0 to all other mappings.

We then use P_j^i to set the maximum values of the zero λ s. For each λ_{Oj}^{Oi} , we will make its maximum value inversely proportional to P_j^i . The less likely it is that a particular mapping is not the primary mapping of some output role, the more likely it is that the mapping will be driven to zero.

The zero λ s will then determine the value of the other λ s. The maximum value of each other λ_{jl}^{ik} will be the minimum of $(1 - \lambda_{Ol}^{Ok}), (1 - \lambda_{Oj}^{Oi})^2$. Hence, primary mappings will mostly strongly

² Where 1 is the maximum value of the λ s.

transfer knowledge to other primary mappings. The likelihood of transferring knowledge from one mapping to another is inversely related to the strength of the force driving either of the mappings to zero.

- iv. We now need an error-minimization procedure which incorporates the method for setting the λ s. We propose to combine error-minimization and systematicity in two steps.

An epoch begins with the presentation of all training items. Gradient descent using the delta rule is performed, and a weight step is taken following this gradient. Let the decrease in error due to this step be called ϵ .

The second step involves using a systematicity bias as a search heuristic for minimizing error. We find the most systematic solution which minimally increases error. First, we assign a λ_{\max} for each λ following the procedure above. We then iteratively perform a line search on all the λ s. For each λ , we find the largest value (between 0 and λ_{\max}) which does not increase error more than $\epsilon / (\text{number of } \lambda_{s+1})$. After all λ s are line searched, the procedure repeats.

- v. Our constraint on the λ optimization means that we will always decrease error. Even if each λ optimization increased error by the maximum amount, total error will still have decreased by $\epsilon / (\text{number of } \lambda_{s+1})$.
- vi. A further consequence of this search method is that it will allow us to generalize to unseen dimensions. If we never see any training examples in a given input role, than transferring mappings to that role will not increase error. λ s pertaining to this untrained input role will therefore be as large as possible, as we are searching for the largest λ which does not increase error.

Furthermore, as long as P assigns a non-zero probability to a least one role-role mapping for each input role, we are guaranteed to get transfer of mappings. Also, since at least one role-role mapping will not move strongly towards zero, the upper limit on transfer to the untrained role will also be set high.

This highlights how the search method implements a bias towards systematicity. The search algorithm is presupposing that this there will be significant transfer between different input role to output role mappings, and will follow this assumption unless there is evidence to the contrary.

- vii. Of course, this network does not have to be systematic. If there are no good correspondences between different role \rightarrow role mappings, then the λ s will

remain at very low values in order to minimize error. Thus, the network will take advantage of systematicity but is not bound to it.

D. A simple localist test of the algorithm on a quasi-systematic training set

i. The algorithm was tested using a small example.

a. Network structure:

Two layers

Input and output representations were identical, consisting of Tensor Products.

Roles	\mathbf{r}^1 [1 0 0 0]	Fillers \mathbf{f}^A [1 0]
	\mathbf{r}^2 [0 1 0 0]	\mathbf{f}^B [0 1]
	\mathbf{r}^3 [0 0 1 0]	
	\mathbf{r}^4 [0 0 0 1]	

b. Training set

Training for each role

Role #	Systematic?	Fillers Trained
1	Yes	A, B
2	Yes	A, B
3	<u>No</u>	A, B
4	Yes*	<u>A only</u>

*The evidence for Role 4 is consistent with roles 1 and 2, but not conclusive as there is no training data on filler B's mapping.

TENSOR PRODUCT		VECTOR	
INPUT	OUTPUT	INPUT	OUTPUT
Identity map for all fillers in roles 1 and 2			
$\mathbf{f}^A \otimes \mathbf{r}^1$	$\mathbf{f}^A \otimes \mathbf{r}^1$	1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0
$\mathbf{f}^B \otimes \mathbf{r}^1$	$\mathbf{f}^B \otimes \mathbf{r}^1$	0 1 0 0 0 0 0 0	0 1 0 0 0 0 0 0
$\mathbf{f}^A \otimes \mathbf{r}^2$	$\mathbf{f}^A \otimes \mathbf{r}^2$	0 0 1 0 0 0 0 0	0 0 1 0 0 0 0 0
$\mathbf{f}^B \otimes \mathbf{r}^2$	$\mathbf{f}^B \otimes \mathbf{r}^2$	0 0 0 1 0 0 0 0	0 0 0 1 0 0 0 0
Each filler maps to the other in role 3			
$\mathbf{f}^A \otimes \mathbf{r}^3$	$\mathbf{f}^B \otimes \mathbf{r}^3$	0 0 0 0 1 0 0 0	0 0 0 0 0 1 0 0
$\mathbf{f}^B \otimes \mathbf{r}^3$	$\mathbf{f}^A \otimes \mathbf{r}^3$	0 0 0 0 0 1 0 0	0 0 0 0 1 0 0 0
Identity map for filler A in position 4			
$\mathbf{f}^A \otimes \mathbf{r}^4$	$\mathbf{f}^A \otimes \mathbf{r}^4$	0 0 0 0 0 0 1 0	0 0 0 0 0 0 1 0

c. Training assumptions

Network is told it's a 1-1 mapping: P assigns a probability of 1 to the most likely mapping and a probability of 0 to all others.

The most likely mapping is determined by distance of each filler-filler mapping from the zero mapping (i.e. a matrix of all zeros).

d. Questions

(1) Can the network learn the mapping?

(2) What is the network's response to untrained $\mathbf{f}^B \otimes \mathbf{r}^4$?

–responding to an input on the untrained 8th input unit

ii. What does the network have to accomplish?

a. With localist role vectors, the problem reduces to simple weight copying; the network must learn to make the value on the unit 8 \rightarrow unit 8 weight approach that of the other unit $j \rightarrow$ unit j weights.

b. Of course, the network is given conflicting evidence about how to map filler B onto the output. But the network can take advantage of the fact that its mapping for filler A is consistent with that of roles 1 and 2. Based on this consistency, the systematicity bias allows the network to assume that its mapping for filler B should be the same as the other two roles.

iii. Results

a. Can the network learn the mapping?

Yes; with a weight step of .3 and a λ maximum of .075, the network converged to an average mean squared error of .000828 in 10 iterations.

b. What is the network's response to untrained $\mathbf{f}^B \otimes \mathbf{r}^4$?

The network responds with the systematic generalization
 $(.67) \mathbf{f}^B \otimes \mathbf{r}^4 + (.0025) \mathbf{f}^A \otimes \mathbf{r}^4$

E. Generalization of algorithm to the distributed role vector case

i. Different training regimen

a. Roles \mathbf{r}^1 [1 -1 -1 -1] Fillers \mathbf{f}^A [1 0]

\mathbf{r}^2 [-1 1 -1 -1] \mathbf{f}^B [0 1]

\mathbf{r}^3 [-1 -1 1 -1]

\mathbf{r}^4 [-1 -1 -1 1]

b. Training set: Same as above with different roles

TENSOR PRODUCT		VECTOR	
INPUT	OUTPUT	INPUT	OUTPUT
Identity map for all fillers in roles 1 and 2			
$\mathbf{f}^A \otimes \mathbf{r}^1$	$\mathbf{f}^A \otimes \mathbf{r}^1$	1 -1 -1 -1 0 0 0 0	1 -1 -1 -1 0 0 0 0
$\mathbf{f}^B \otimes \mathbf{r}^1$	$\mathbf{f}^B \otimes \mathbf{r}^1$	0 0 0 0 1 -1 -1 -1	0 0 0 0 1 -1 -1 -1
$\mathbf{f}^A \otimes \mathbf{r}^2$	$\mathbf{f}^A \otimes \mathbf{r}^2$	-1 1 -1 -1 0 0 0 0	-1 1 -1 -1 0 0 0 0
$\mathbf{f}^B \otimes \mathbf{r}^2$	$\mathbf{f}^B \otimes \mathbf{r}^2$	0 0 0 0 -1 1 -1 -1	0 0 0 0 -1 1 -1 -1

Each filler maps to the other in role 3

$$\begin{array}{rcl}
 \mathbf{f}^A \otimes \mathbf{r}^3 & \mathbf{f}^B \otimes \mathbf{r}^3 & -1 -1 1 -1 0 0 0 0 \quad 0 0 0 0 -1 -1 1 -1 \\
 \mathbf{f}^B \otimes \mathbf{r}^3 & \mathbf{f}^A \otimes \mathbf{r}^3 & 0 0 0 0 -1 -1 1 -1 \quad -1 -1 1 -1 0 0 0 0 \\
 \\
 \text{Identity map for filler A in position 4} \\
 \mathbf{f}^A \otimes \mathbf{r}^4 & \mathbf{f}^A \otimes \mathbf{r}^4 & -1 -1 -1 1 0 0 0 0 \quad -1 -1 -1 1 0 0 0 0
 \end{array}$$

ii. This highlights the abstract nature of the generalization pressure; here, we are not simply copying weights (as no unit represents only one role).

iii. Results

a. Can the network learn the mapping?

Yes; with a weight step of .1 and a λ maximum of .075, the network converged to an average mean squared error of .009046 in 5 iterations.

b. What is the network's response to untrained $\mathbf{f}^B \otimes \mathbf{r}^4$?

The network responds with the systematic generalization.

$$(.47) \mathbf{f}^B \otimes \mathbf{r}^4 + (.0017) \mathbf{f}^A \otimes \mathbf{r}^4$$

F. Conclusion

Have recognized the need for structure-sensitive learning biases, we have shown how such biases could be incorporated into a network operating over tensor product representations. The method sketched here shows how these structure-sensitive biases will lead to systematicity of mapping, transferring knowledge about fillers to roles in which they have not been seen.

References

Kohonen, T. (1977). Associative memory: A system-theoretical approach. Berlin: Springer-Verlag.

Marcus, G. (1998). Can connectionism save constructivism? Cognition, 66, 153-182.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. Artificial Intelligence, 46, 159-216.

Smolensky, P. (1995). Constituent structure and explanation in an integrated connectionist/symbolic cognitive architecture. In C. Macdonald & G. Macdonald (Eds.). Connectionism: Debates on Psychological Explanation, Volume Two, pp. 221-290. Oxford: Basil Blackwell.

Department of Cognitive Science
 Johns Hopkins University
 3400 N. Charles St.
 Baltimore, MD 21218
 USA

{goldrick, hale, mathis, paul}@mail.cog.jhu.edu