Philosophical Psychology
Vol. 18, No. 3, June 2005, pp. 313–325

Routledge
Taylor & Francis Group

Atomistic Learning in Non-Modular Systems

James Blackmon, David Byrd, Robert Cummins, Pierre Poirier & Martin Roth

We argue that atomistic learning—learning that requires training only on a novel item to be learned—is problematic for networks in which every weight is available for change in every learning situation. This is potentially significant because atomistic learning appears to be commonplace in humans and most non-human animals. We briefly review various proposed fixes, concluding that the most promising strategy to date involves training on pseudo-patterns along with novel items, a form of learning that is not strictly atomistic, but which looks very much like it 'from the outside'.

1. The Problem

The assumption that we have the ability to evaluate and add or delete beliefs individually is common in the psychological literature on memory, concept acquisition, and language acquisition (Ramsey, Stitch, & Garon, 1990; Cummins, Poirier, & Roth, 2004). Indeed, this supposition pervades our informal, common-sense framework for understanding the mind, as well as our formal-symbolic models of rationality and epistemology. Rationality, for example, is thought of as at least in part about the management of beliefs and other propositional attitudes: what beliefs we should and should not adopt, when and what to add, when and what to delete, etc. Models of rationality tell us when we *ought* to revise our individual beliefs, and because *ought* implies *can*, these models presuppose that we *can* manage our beliefs individually.[1]

At the same time, most of our current connectionist models of cognition suggest that the knowledge used in carrying out many cognitive tasks related to memory,

Correspondence to: David Byrd. Email: davidbyrd101@hotmail.com

ISSN 0951-5089 (print)/ISSN 1465-394X (online)/05/030313–13 © 2005 Taylor & Francis Group Ltd
DOI: 10.1080/09515080500177283

learning, forgetting, and concept acquisition is fully distributed. As we will show, there are fairly strong reasons for thinking that current connectionist models in which knowledge is fully distributed cannot reliably add or delete beliefs individually. Instead, they have a strong tendency to obliterate previously stored beliefs when a new belief is introduced. As a result, either these models need to be changed to accommodate the pervasive assumption of individual belief management, or the underlying assumption needs to be modified.

Unfortunately, as it stands the question is too under-described to be answered properly. One constraint that a system must satisfy in order to be what we will call an *atomistic* learner is that the *learning mechanism* guarantees that previous information is conserved in the process of acquiring new atoms of information.[2] For example, here is a way to get a connectionist system trained by back-propagation to add an atom of information that would *not* count as atomistic learning: train a system on training set items {*a*, *b*, *c*} so that it learns to respond to each correctly. Once the system performs on this initial set at an acceptable level, randomize the weights and train the system on the union of the initial training set {*a*, *b*, *c*} and the new item {*d*}, and stop training when the system performs adequately on all four items. If this method, which we call *retro-training*, counted as atomistic learning, connectionist systems in which knowledge is fully distributed would evidently have the capacity for atomistic learning and forgetting.

However, notice that this method of learning begins by obliterating all the previous information when the weights are randomized. If all previous information is obliterated in favor of the new information, then it is only a matter of accident or experimental design, and not a consequence of the learning mechanism itself, that the system after training bears an interesting epistemic relation to the system before retraining, viz. handles {*a*, *b*, *c*} plus one new atom {*d*}. Something somewhere must reintroduce the initial training set. By contrast, atomistic learning requires the ability to add information, while conserving previous information, without re-training on the initial set.

This is the important sense in which new information can be added or deleted *individually*. For example, medical students take many years of learning to associate anatomical structures with their Latin names. It is prima facie plausible to suppose that when a medical doctor has to learn the name of a new structure later in life, she can simply add a new name–structure pair, while conserving previous information, *without* retro-training. And while acquiring I/O pairs is not equivalent to acquiring beliefs, if a connectionist network in which knowledge is fully distributed cannot reliably acquire *I/O pairs* atomistically, then surely acquiring *beliefs* atomistically is not on the cards, either. So although the focus of the rest of the paper is on I/O pairs and whether networks in which knowledge is fully distributed can learn them atomistically, the consequences of our results for a connectionist story about atomistic belief management should be obvious. So our question becomes: do connectionist systems in which knowledge is fully distributed have the capacity for *atomistic* learning (and forgetting), that is, learning that conserves previous knowledge but does not rely on previous training sets?

2. The Argument

In this section, we argue that atomistic learning/forgetting is problematic in non-modular networks, i.e., networks that are not collections of largely independent networks connected or gated together in some highly specialized way. We begin with two preliminaries. First, we stipulate some terminology. Then we hedge by specifying some things that we are NOT claiming.

2.1. Terminology

A. Novel case learning

At a moment, i.e., ignoring weight changes, we can regard a network as a function from a vector and a weight matrix to an output vector.[3] The input–output function thus typically changes over time as the result of alterations in the weights. (It might also change as the result of changes in basic architecture, e.g., the addition or subtraction of nodes or alterations in activation functions.[4] We will ignore these for the present.) Such changes are regarded as learning or forgetting when *desired* input–output pairs are added or lost.

It is important to keep in mind that, in typical network models, most inputs are of no interest because they do not encode anything pertinent to the task. Thus, for a simple XOR net, the inputs of interest are (1,1), (1,0), (0,1), and (0,0), but the network will respond to any real-valued vector in the range of activation. Changes in responses to vectors other than the four just listed, however, are not regarded as learning or forgetting. This is reflected in the fact that we cannot assess error in such cases: there is no correct (or better) response to $(0.32, 0.75)$. Input vectors that have an interpretation in the domain will be called *significant.*

Output vectors present a different case: every output vector resulting from a meaningful input vector is considered significant in the sense that its distance from a correct response—its error—can be measured. Hence, any change in the output generated by a given significant input may be regarded as learning (less error) or forgetting (more error).

Suppose we wish to train a network to respond to an encoding of US state names with an encoding of the names of their capitals. As the training proceeds, we should see improvement on the pairs in the training set. We will also see some generalization, but it will be pure luck if any state-capital pairs not in the training set are learned, since the domain is not systematically structured. In general, there will be some pairs not in the original training set that the network gets wrong no matter how long we train on that training set. Learning a new pair in an unstructured domain (that is, one in which associations are arbitrary and not generalizable) like this is an example of what we call *novel case* learning. Novel case learning contrasts with both improvement on familiar cases, and with generalization effects in which there is improvement on significant inputs not in the training set during the course of training on that same set. In these latter cases, the added knowledge is not totally novel.

A more interesting case of novel case learning is the mastery of exceptions in an otherwise highly structured domain, e.g., learning that the past tense of 'go' is 'went' rather than 'goed'. As we will see shortly, what makes these cases interesting for present purposes is that there is a trade-off between the mastery of novel cases and the generalization effects that are essential to learning in structured domains.

B. Atomistic learning

Novel case learning will be said to be atomistic when a novel case I/O pair can be added to the system's repertoire without retro-training.

C. Non-modular

Atomistic learning is clearly possible in highly modular systems in which, essentially, novel cases are acquired by training a set of connections not involved in previously learned pairs. The interesting question concerns whether non-modular networks are capable of atomistic learning. A network will be called non-modular if every weight is potentially open to modification (within the same dynamic range) during learning. We require only potential openness to modification in order to accommodate cases in which particular weights might in fact require no modification during learning, but in which nothing about the architecture or learning algorithm affords any in-principle reason why those particular weights should not be involved or even critical. The rationale underlying this formulation should become clear as we proceed.

It should be apparent that networks that are non-modular in our sense are networks in which information stored in the weights is distributed in the following sense: nothing about the network architecture or learning algorithm constrains any particular connection or set of connections to be involved exclusively in the pairing of some proprietary input–output pairs.[5]

2.2. What we are not *claiming*

We are not claiming that atomistic learning is impossible in non-modular networks. Our claim is, rather, that there is no principled mechanism for guaranteeing its success, particularly in unstructured domains, or structured domains involving significant 'exceptions'. Atomistic learning thus contrasts with such features as spontaneous generalization and graceful degradation. These properties of non-modular networks are automatic consequences of the architecture and must be explicitly suppressed if they are not desired for some reason. In the case of atomistic learning, on the other hand, overwriting of previous learning is the default, as it were: it will happen unless special measures are taken to prevent it or conditions are particularly serendipitous.

2.3. What we are *claiming*

When it does occur in non-modular networks, atomistic learning is essentially serendipitous. Intuitively, the idea is quite simple: adding enough novel cases, and training long enough on *them alone* will eventually overwrite the effects of

previous training. This will happen even though there is 'plenty of space' in the sense that, were one to train on the old and novel items together (or a representative sample of them), the novel items would be learned without detriment to the consequences of previous learning. We will express this last point by saying that the novel case learning problem *has a solution*, meaning that there is a point in the weight space of the net that preserves the desired results of previous training (including desired generalizations) while also handling the novel cases. Notice that a novel case learning problem might have a solution even though that solution cannot be found atomistically—i.e., by training on the novel cases alone. Without the proviso that the novel case learning problem has a solution, our central claim becomes trivial, since it is evidently always possible to add so many novel cases that the network will become saturated simply by learning *them*.

Now consider a case in which a novel case learning problem L has a solution. Call a point w' in a weight space W conservative for L relative to another point w if w' is a solution to L for a network that finds itself in w. Call a point w' non-conservative for L relative to w if a change from w to w' accommodates the novel cases at the cost of significant damage to the results of previous learning. For a given novel case learning problem, there will, in general, be many points w' in W that are non-conservative for L relative to w. If the network is trained on the novel cases alone, there is no reason to assume that a point that is conservative for L relative to w will be found rather than one that is not, since the learning rule will treat w in the same way it treats the initial random weights in any *de novo* learning problem: all are equally open to revision, including those that supported the correct responses to previously learned items.

The weights at w are most likely to generate what the new training regime will take to be errors if the novel case inputs are geometrically close to one of the previously trained inputs. In fact, we can guarantee overwriting of any previously learned input–output pair (x, y) by training on a sufficiently large class of new cases in which the inputs are close to x but the required outputs are all mapped onto an output y' distant from y.[6] In such a case, the network will typically respond initially to the novel cases with an output similar to y. This is just the definition of generalization. These responses will generate large error signals. As training proceeds, the set of similar novel inputs will eventually be mapped onto y'. Since the novel inputs are no more similar to each other than any one of them is to x, generalization will ensure that x is mapped onto y' as well. We can, as it were, surround x with bad company that will eventually corrupt it. (See Figure 1.)

It is tempting to respond to this construction by pointing out that overwriting need not occur when novel case inputs are not similar to old-case inputs. This is true but beside the point. In general, the network cannot anticipate what sorts of new inputs it will encounter. The only way to prevent overwriting scenarios like the one described above in atomistic learning is to guarantee that novel case inputs are not similar enough to any previously learned inputs to produce 'generalization drag'. But such a solution would be both draconian and ad hoc. It would be draconian, because it would block significant generalization effects across the board.[7] It would be ad hoc, since natural input encodings would have to be transformed in a way that keeps

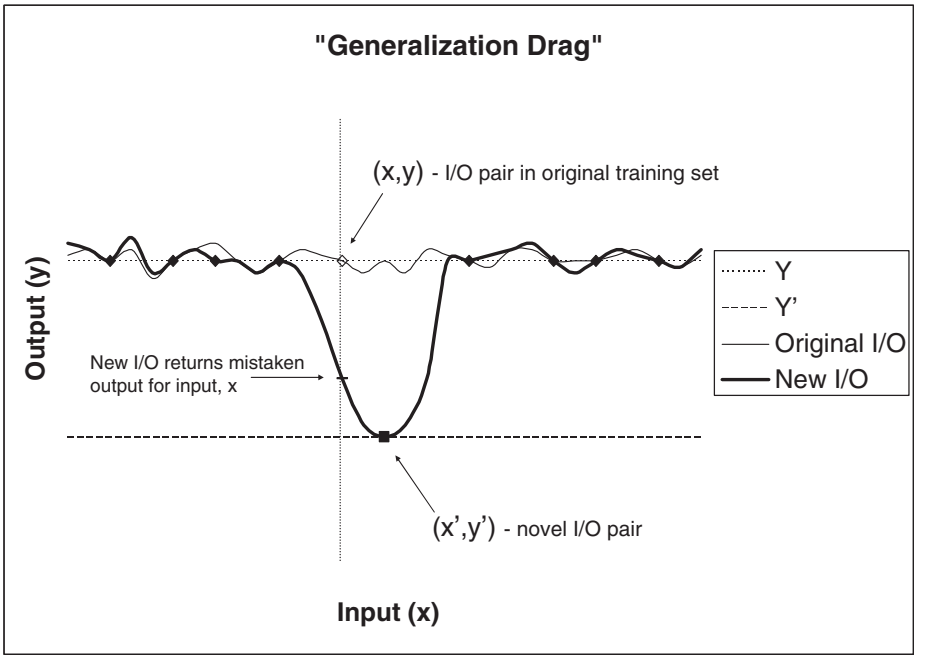

Figure 1 Generalization drag.

similarities between input vectors proportional to the similarities between their corresponding outputs. No natural system could hope to meet such a condition, since it would require foreknowledge of the correct input–output parings, which is exactly what the network is supposed to *learn*. Nature is bound to present us with some 'hard' learning problems, i.e., learning problems in which similar inputs sometimes require dissimilar outputs.

Of course, domains do not come neatly divided in to structured and unstructured; rather, the degree of structuring varies from extreme cases such as the state capitals' case at the unstructured end of the continuum, to cases such as geometrical sequences or regular pluralization or tensing of verbs at the structured end. Learners cannot expect to know in advance how structured a domain is, or where exceptions are likely to occur.

3. The Literature

In light of these difficulties with atomistic learning, it may be tempting to simply abandon non-modular networks in favor of modular ones. While some have favored this solution,[8] many find abandoning non-modularity too high a price to pay. Recall that two desirable properties that fall out automatically from non-modular networks are spontaneous generalization and graceful degradation (robustness), and these are compromised when we give up non-modularity. Consequently, it has been hoped

that a solution to the atomistic learning problem can be found that does not require abandoning these two attractive features of non-modular networks.

What we call the problem of atomistic learning and forgetting is an instance of a more general problem widely discussed in the connectionist modeling literature: the problem of catastrophic interference. First studied systematically by McCloskey and Cohen (1989) and Ratcliff (1990), the problem of catastrophic interference is just the phenomenon that new learning in a connectionist network tends to overwrite previously learned knowledge. In the two articles just cited, many attempts were made to reduce the interference of new and old knowledge by varying parameters, such as the number of hidden units and the learning rate, by massively over-training on previously learned knowledge before training on the new knowledge, by using local representations for the stimuli and responses, and by freezing during the acquisition of new knowledge the weights most involved with previously acquired knowledge. Nothing worked. Though it is not surprising that many of these attempts failed, the failure of the last attempt may seem somewhat puzzling. McCloskey and Cohen (1989) have a nice explanation of the failure of the 'freezing' technique:

> At first glance, this result might seem nonsensical. If the weights established by A-B learning were fixed during A-C training, how could performance on the A-B list have been disrupted? The answer is that the weights established by the A-C training, as well as those resulting from A-B training, come into play whenever an input is presented to the network. Thus, the combined effect of the weights established by A-B training and the weights established during A-C training was to produce incorrect patterns of activation on the output units when the A-B list was tested after some A-C training. (1989, p. 140)

More recently, French (1992) suggested using a 'node sharpening' learning algorithm to produce sparse encodings at the hidden layer. One problem for this strategy is that the more 'atomistic' learning is facilitated by reducing hidden layer overlap, the less the network is capable of generalizing, a point we alluded to at the end of the previous section. This is, perhaps, desirable in unstructured domains, but not desirable in structured domains containing 'exceptions', for the absence of generalization cripples both the rate and ability of networks to acquire knowledge of structured domains. This point also holds for strategies to avoid catastrophic interference by 'orthogonalizing' the inputs. Interference is avoided by insuring the complete dissimilarity of inputs from one another, but only at the expense of generalization effects. Since generalization effects are just the production of similar outputs for similar inputs, such effects would be precluded by requiring the inputs to be completely dissimilar from one another. Further, more 'fine-tuned' attempts to orthogonalize only the inputs of I/O pairs likely to bring about catastrophic interference would be unrealistic since the network cannot differentially act on the effects of training that has not yet taken place.

One recent suggestion that seems to evade both the loss of generalization and any reliance on stored training sets involves rehearsal over *pseudo-patterns*. First formulated by Robbins (1995) and later incorporated into a dual memory model,[9] the method of rehearsal on pseudo-patterns approaches being functionally equivalent

to retro-training on previously learned knowledge (with respect to the changes it brings about in the I–O function) but without needing access to the input–output patterns of earlier training sets. The proponents of the latest version of this method (Ans, Rousset, French, & Musca, 2002) present a 'reverberating' simple recurrent network (RSRN) that can successfully learn new 'temporal sequences of patterns' without catastrophically forgetting previously computable sequences.[10] The RSRN's success is attributed to its use of 'reverberated' *pseudo-patterns* (or *pseudo-episodes*, due to Robbins) which are generated by a separate network which, together with the first network, forms a dual-network architecture.[11]

A pseudo-pattern is an input–output pair where the input is internally-generated noise and the output is whatever the weight matrix yields given that noise as input. Pseudo-patterns that result from reverberation are held to be particularly *representative* of the entire input–output function that the network computes. Such a pseudo-pattern is, in a crucial sense, entirely independent of the specific training set that caused the network to obtain whatever weight matrix it has. No record or interleaving of the training set is used.[12] Nor is there any storing or interleaving of anything that represents that particular training set (as is the case with exemplars). All that gets 'stored' and interleaved here during successive training episodes are these representatives of the RSRN's prior computational capacity. Furthermore, these can be generated 'on the fly' and only need to be stored temporarily from the beginning of a new training episode to the end. Because retraining for an RSRN needs no explicit access to the previous set it appears to be a more plausible and efficient model.

So far, the results for RSRNs are impressive. It should be noted, however, that the RSRN approach involves training on more than the new items. The network doing the learning is trained on the new items together with the pseudo-patterns generated by its 'clone'. This is evidently a huge improvement on retro-training, since the required pseudo-patterns can be generated on the fly when new learning occurs; there is no need to store previous training sets, representative subsets of them, or exemplarized versions of them. This is, at least, a case of individual belief management that is far more plausible than comparable effects achieved by retro-training and its kin.

Two special features of RSRNs would benefit from independent motivation from the neurosciences: the employment of 'autoassociative' nodes and the notion of 'reverberation'. Autoassociative nodes comprise a proper part of the output layer of the RSRN. They simply compute the identity function for the input, effectively giving any system upstream (such as a teacher program) access to exactly what was given to the RSRN as input. Reverberation involves computing the error between the autoassociative output (which is equal to the input) and the network output, then using back-propagation on this (in addition to any error calculated in the standard way, that is, as the difference between output and target values) to make weight changes. This process is repeated a number of times.[13] Reverberated pseudo-patterns are found to be more effective than normal pseudo-patterns at avoiding catastrophic interference.

Independent support from the neurosciences is needed to demonstrate that these two features of RSRNs are more than simply shrewd contrivances. Enthusiasm resulting from the achievements of such models is thus premature, in light of the fact that their neurophysiological plausibility remains to be established.

4. Desirability of a General Solution

Given the above, the most a non-modular network can hope for is to attempt to mitigate the near impossibility of atomistic learning. Such an attempt was made by McClelland, McNaughton and O'Reilly (1995), where it was suggested that particular learning episodes might be quickly accommodated by the hippocampus, and that this new knowledge might then be slowly integrated into the corpus of previous knowledge stored in the neocortex. According to this model, catastrophic interference would be avoided, or at least lessened, if the neocortex was trained not only on the new knowledge, but on the new knowledge 'interleaved' with exemplars of previous knowledge.

The first thing to note about this technique is that such 'interleaving' is nothing more than what we call 'retro-training', with the exception that what is added to new knowledge is not everything the net has previously learned, but rather only highly representative examples of previous knowledge. What this method leaves unexplained is how such exemplars are created and stored. Since it is unlikely the environment would continually produce exemplars of all of one's previous knowledge, created exemplars would have to be stored in the hippocampus, an area, according to McClelland et al., associated with short-rather than long-term memory.[14] Further, since this strategy relies on the use of highly representative examples from structured domains, this approach does not seem to successfully account for the acquisition of items from structured domains *with exceptions*, or unstructured domains, e.g., a set of name-to-phone-number pairs. The reason is that such domains lack the kind of common structure among items that would yield an exemplar. And without an exemplar, the current learning model collapses to having to retrain over all previously learned name-to-phone-number pairs to prevent catastrophic interference.

There thus appear to be two ways one can try to deal with atomistic learning in connectionist networks. One way is to cleave to non-modular networks and attempt to mitigate the overwriting problem in some way, e.g., by enforcing sparseness at the hidden layer, and paying the consequent cost. Or, one can abandon the attempt to model atomistic learning in non-modular networks and focus instead on psychologically and neuroscientifically plausible strategies for training on more than the novel cases alone. To date, the RSRNs alluded to above seem the most promising version of this strategy, because only few pseudo-patterns need to be added to the novel case for training, and these pseudo-patterns can be generated without access to any previously learned I/O pairs simply by 'reverberating' a random input through the network until its behavior stabilizes. We suspect that RSRNs might account naturally for a number of otherwise puzzling learning phenomena as well.[15]

5. Conclusion

When non-modular connectionist networks were initially proposed, it was well understood that they were idealized in several ways. The standard connectionist account of cognition idealizes away from the role of neurotransmitters, the shapes of spike trains, the non-local effects of a chemical diffusing over an entire population of neurons,[16] and many other potentially important properties of actual neural networks. The idea is that it is, perhaps, best to begin with a simple and well-understood model, adding complications only when forced to do so to accommodate otherwise recalcitrant phenomena. In that way, we are more likely to have some idea of what the extra complexity is actually for—what it makes possible. We suspect that the appearance of atomistic learning is such a phenomenon, for it seems to force us to tolerate some modularity—perhaps in the form of a pseudo-pattern extractor that is distinct from the network that stores the knowledge, or perhaps in the form of more radical changes in our understanding of neural computation at the cellular or synaptic level.

For those attached to standard non-modular connectionist modeling, it is tempting to attempt to undercut the offending problem: perhaps we are rather bad at learning unstructured domains, and at learning exceptions in highly structured domains. We are somewhat sympathetic to this response: tasks that are problematic for us ought to be problematic for our best models. But with great success we can and do learn phone numbers, the state capitals, an enormous amount of vocabulary in first and second languages, and irregular verbs in first languages. So it cannot plausibly be held that these network deficiencies are reflective of our own cognitive shortcomings.

Another objection we have heard is that beliefs simply are not added or deleted one at a time, since changing any belief typically involves a cascade of changes in other beliefs. While this might be doubted—a new phone number, a new vocabulary word, a switch from 'goed' to 'went' do not produce cascades in the way discovering the finite velocity of light did[17]—we are prepared to concede the point because it is beside the point. Revision of one's beliefs in the light of new information seems to require something like atomistic belief management. It certainly is not compatible with the kind of catastrophic and indiscriminate degradation of everything previously learned that standard networks suffer when trained solely on a novel I/O pair.

The pervasiveness of the assumption that we are able to manage our beliefs atomistically can hardly be denied. All standard epistemology, for instance, simply presupposes that we have this capacity. Now standard epistemology might, in the end, turn out to be inapplicable to actual minds,[18] but suffice it to say that abandonment of this supposition is not to be taken lightly. And though atomistic belief management is not to be modeled by the independent addition or deletion of learned I/O pairs, we do hold that its accommodation *requires* a solution to the specific problem we have highlighted *if* systems that have such a capacity are to be modeled as connectionist networks. For, if a system cannot add I/O pairs

atomistically, or achieve a reasonable substitute for it, doing anything approaching atomistic belief management seems hopeless.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0137255. We would like to acknowledge Chris May, Alexa Lee, Robert French, Jim Garson, and Paul Teller for helpful comments and suggestions on earlier drafts of this paper.

Notes

[1] Bayesian models, for instance, assume we can assign probabilities to individual beliefs and modify those probabilities individually as a function of current evidence.

[2] Of course, given finite memory, there will always be a point where a system can no longer add new information without deleting old information. So this point only applies to cases where there is enough room in memory to add the new information in the first place. Also, this constraint is not intended to rule out cases where, upon learning an atom, an inference rule results in updating (and therefore possibly *not conserving*) other relevant beliefs.

[3] Networks can, of course, be stochastic, in which case there need not be a unique output on a given input. In multi-layer feed-forward networks, output is $W_n \times W_{n-1} \times \cdots \times W_1 \times \mathbf{I}$, where $\mathbf{I}$ is the input vector, and W_x is the weight matrix connecting layer x and layer $x+1$. In recurrent nets, the process that generates an output vector from an input vector and the weights is more complex and varies with the architecture. Nevertheless, given a fixed architecture, the input and the weights suffice to fix the output vector (modulo whatever randomness there is in the system).

[4] Any change in input–output behavior that can be effected by weight changes can be effected by fixing the weights and altering the activation functions of the individual units. In some cases, learning not only affects weights, but also effects changes in activation functions, e.g., by altering thresholds.

[5] The sense in which a *network* is said to be distributed (i.e., non-modular) should be distinguished from the sense in which its activation vectors are said to use a distributed encoding scheme. An encoding scheme for a vector $\mathbf{v}$ is fully distributed relative to a domain of representational targets D if every element of $\mathbf{v}$ is involved in the representation of every element of D. A fully distributed *network* can utilize a local encoding scheme, and a local network can utilize a distributed encoding scheme.

[6] In this case, the whole class is novel with respect to previous learning, in that the new pairs are not accommodated by generalization from previously learned cases. The pairs in the class are not 'novel' with respect to each other, since, owing to their similarity to each other, training on a few will improve performance on others via generalization. In the latter case, we have a whole class of 'exception' cases with inputs similar to a previously learned input.

[7] In a strictly unstructured domain, preventing generalization might be desirable. But cortical circuits cannot be expected to know in advance whether the domains they are recruited to learn are structured or not. Moreover, as we emphasized earlier, an important case of novel case learning involves mastery of exception cases in otherwise highly structured domains. Blocking generalization in these cases would evidently be catastrophic unless it could be blocked only around the exceptions. But this would evidently require supernatural prescience.

[8] For example, Page (2000). Page thinks that localist solutions can preserve generalization and graceful degradation. However, we are skeptical. First, the kind of generalization that he is talking about is not the kind of generalization that we describe here. Rather, it is the kind of

generalization that one finds in Rumelhart and McClelland's (1986) Jets and Sharks network: given the activation of a node that represents a Shark (or Jet), several other nodes that represent various properties of a typical Shark (or Jet) will be activated. This is a very different sort of generalization from the one discussed here, where what is relevant is similarity of response given similarity of input. Second, it is unclear whether Page's networks are local in the sense in which we use the term. He seems to be concerned with the issue of localism at the level of representation in the activation vectors, while we are discussing localism in the sense of modularity.

[9] French, 1997; Ans and Rousset, 1997; French, Ans, & Rousset, 2001; Ans et al., 2002.

[10] This is an improvement on previous work in which catastrophic forgetting was avoided for networks that trained on 'a series of, ''static'' (non-temporal) patterns' but not for temporal patterns (Ans & Rousset, 1997; French, 1997; Robbins, 1995). Though these earlier networks would suffice to illustrate the learning of new I/O pairs without catastrophic interference, we use the RSRN network because it represents the latest, most powerful incarnation of the pseudo-pattern strategy.

[11] Dual-network architectures are not without neurological and theoretical precedent. See McClelland et al. (1995) for an account of such architectures in terms of the observed limitations and successes of connectionist networks. We present a brief summary in the next section.

[12] Even the proponents of pseudo-pattern rehearsal seem to understate this point. They write about pseudo-patterns reflecting what has been 'previously learned' as if the pseudo-pattern encodes which aspects of the resulting weight matrix (or of the function it computes) were due to a learning episode as opposed to being already there. But from what they tell us about pseudo-patterns, RSRNs do not use such a distinction.

[13] Ans et al., 2002.

[14] Incidentally, the RSRN approach of using pseudo-patterns seems to solve both the creation and storage of such representative knowledge.

[15] Consider intrusions, i.e., spontaneously generated but robust errors. These are difficult to explain in standard network models, because error signals during learning are generated only by comparison with correct responses. Really robust errors that are not simply the result of limited capacity have to be explicitly trained, and are therefore not spontaneous but due to existing errors in the training set. RSRNs, however, are trained on pseudo-patterns. Training on pseudo-patterns, as opposed to explicit retro-training, conserves everything equally: previously acquired responses to non-significant inputs are on all fours with previously learned responses to significant inputs. The process that preserves previous learning is blind to the distinction between inputs that are significant and those that are not. A lot of 'trash' is preserved. Preservation of 'trash' pairs induces generalization effects in exactly the same way that preservation of significant pairs does, and this can be expected to introduce intrusions when 'trash' inputs are sufficiently close to significant ones. Moreover, errors, once introduced, will tend to be preserved unless they are explicitly overwritten.

[16] For this last property, see Husband, Smith, Jakobi and O'shea (1998).

[17] Of course, other beliefs may seem to 'come along for the ride', but here we must recognize those additional beliefs that have some epistemic justification. For example, upon learning what my aunt's phone number is I may form beliefs such as *that the prefix of her number is identical to the prefix of my dentist's number*, or *that my aunt has a number that contains no evens*. As we see it, these are beliefs that are inferred (with some epistemic justification) from the original belief which pairs my aunt with a specific number. That other beliefs can be and are inferred from one belief is, on its own, no reason to hold that the one belief is not learned individually.

[18] We have some sympathy with this idea as well. See Cummins et al. (2004).

References

Ans, B., & Rousset, S. (1997). Avoiding catastrophic forgetting by coupling two reverberating neural networks. *Academie des Sciences, Sciences de la vie*, *320*, 989–997.

Ans, B., Rousset, S., French, R., & Musca, S. (2002). Preventing catastrophic interference in multiple-sequence learning using coupled reverberating Elman networks. In *Proceedings of the 24th annual conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum & Associates.

Cummins, R., Blackmon, J., Byrd, D., Poirier, P., Roth, M., & Schwarz, G. (1999). The epistemology of non-symbolic cognition: Atomistic learning and forgetting. *Tech. Report Phil99*–3. Davis, CA: University of California, Davis.

Cummins, R., Poirier, P., & Roth, M. (2004). Epistemological strata and the rules of right reason. *Synthese*, *141*, 287–331.

French, R. (1991). Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks. In *Proceedings of the thirteenth annual cognitive science society conference* (pp. 173–178). Hillsdale, NJ: Lawrence Erlbaum & Associates.

French, R. (1992). Semi-distributed representations and catastrophic forgetting in connectionist networks. *Connection Science*, *4*, 365–377.

French, R. (1997). Pseudo-recurrent connectionist networks: An approach to the ''sensitivity-stability'' dilemma. *Connection Science*, *9*, 353–379.

French, R., Ans, B., & Rousset, S. (2001). Pseudo-patterns and dual-network memory models: Advantages and shortcomings. In R. French & J. Sougne (Eds.), *Connectionist models of learning, development and evolution* (pp. 13–22). London: Springer.

Husband, P., Smith, T., Jakobi, N., & O'Shea, M. (1998). Better living through chemistry: Evolving gas nets for robot control. *Connection Science*, *10*, 185–210.

Kruschke, J. (1992). ALCOVE: An exemplar-based model of category learning. *Psychological Review*, *99*, 22–44.

McClelland, J., McNaughton, B., & O'Reilly, R. (1995). Why there are complimentary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*, 419–457.

McCloskey, M., & Cohen, N. (1989). Catastrophic interference in connectionist networks: The sequential learning Problem. In G. H. Bower (Ed.), *The psychology of learning and motivation*, (Vol. 24, pp. 109–164). New York: Academic Press.

Murre, J. (1992). *Learning and categorization in modular neural networks*. Hillsdale, NJ: Lawrence Erlbaum & Associates.

Page, M. (2000). Connectionist modeling in psychology: A localist manifesto, *Behavioral and Brain Sciences*, *23*, 443–512.

Ramsey, W., Stich, S., & Garon, J. (1990). Connectionism, eliminativism, and the future of folk psychology. In James E. Tomberlin (Ed.), *Action theory and philosophy of mind—philosophical perspectives* (Vol. 4; pp. 499–533). Atascadero, CA: Ridgeview.

Ratcliff, R. (1990). Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review*, *97*, 285–308.

Robbins, A. (1995). Catastrophic forgetting, rehearsal, and pseudorehearsal. *Connection Science*, *7*, 123–146.

Rumelhart, D., & McClelland, J. (1986). On learning the past tenses of English verbs. In D. Rumelhart, J. McClelland, and the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 2; pp. 216–271). Cambridge, MA: The MIT Press.