A Neural Implementation of MINERVA 2

Erik D. Reichle (erik.reichle@mq.edu.au)
School of Psychological Sciences, Macquarie University, NSW 2109, Australia

Aaron Veldre (aaron.veldre@sydney.edu.au)
School of Psychology, The University of Sydney, NSW 2006, Australia

Lili Yu (lili.yu@mq.edu.au)
School of Psychological Sciences, Macquarie University, NSW 2109, Australia

Sally Andrews (sally.andrews@sydney.edu.au)
School of Psychology, The University of Sydney, NSW 2006, Australia

Abstract

The MINERVA 2 (Hintzman, 1984) model of human memory has been used to simulate a variety of cognitive phenomena. These simulations, however, describe cognitive phenomena at Marr's (1982) representation/algorithm level, with little effort to link the core assumptions of the model to an underlying neural implementation (however, see Kelly et al., 2017). This article describes a possible neural implementation of MINERVA 2—one that is simple and arguably biologically plausible. This implementation suggests a novel method for generating response latencies and provides a concrete example to support Marr's claim that the representations and algorithms that mediate human performance in a variety of different cognitive tasks (e.g., decision making; Dougherty, Gettys, & Ogden, 1999) can be investigated and simulated without reference to their underlying neural implementation.

Keywords: connectionist network, MINERVA 2, neural network

Introduction

MINERVA 2 (Hintzman, 1984) is a computer model of human memory that instantiates the core assumptions of an early "resonance" theory of memory that was first proposed by Richard Semon (1923; see Schacter et al., 1978). The core assumptions include the following. First, conscious experiences in *primary* or short-term memory are encoded into *secondary* or long-term memory as discrete *engrams* or memory traces that consist of sets of perceptual and cognitive features representing those experiences. Second, the information represented by these traces can be retrieved by probing secondary memory with a cue that consists of some number of features. This cue will then activate traces to the degree that their contents are similar to the contents of the cue, thereby generating a response from memory called an *echo*. This echo has an *intensity* that increases with the overall familiarity of the cue, thereby allowing previously learned information to be recognized. The echo also contains a composite pattern of features, or *content*, that often includes features not in the cue, allowing previously learned information to be recalled. Finally, because the echo content reflects the global contents of memory, the model provides an account of how abstract categories are learned.

To date, MINERVA 2 has been used to simulate a variety of basic phenomena related to human memory. These demonstrations include category learning and the abstraction of prototypes from specific exemplars (Hintzman, 1986); how changes in the encoding and retrieval contents can produce a failure to recognize words that can be recalled (Hintzman, 1987); and key results from a large number of classic memory experiments (e.g., frequency judgments, list-length effects, levels-of-processing effects, etc.; Hintzman, 1988; see also Collins et al., 2020). Variants of the model have also been used as theoretical frameworks to simulate and explain human behavior across a variety of different task domains, including word identification (Ans et al., 1998; Goldinger, 1998; Kwantes & Mewhort, 1999; Reichle, 2021; Reichle & Perfetti, 2003), sentence processing (Jamieson & Mewhort, 2009), and decision-making heuristics (e.g., availability; Dougherty et al., 1999).

Because MINERVA 2 provides a precise account (summarized below) of human memory and many cognitive tasks that depend upon it, one might ask about the biological plausibility of the model. In other words, are the model's assumptions consistent with what is currently known about how the brain operates? Although the model was originally proposed (Hintzman, 1984) as a functional description of memory (e.g., consistent with Marr's, 1982 representation/ algorithm level) and is agnostic about neural implementation, Hintzman (1990, p. 113, Figure 1e) indicated that MINERVA 2 is equivalent to a single-layered network with connections between a layer of input/output nodes (representing features of the probe and the echo) and a layer of hidden nodes (representing the memory traces of individual experiences). The "features" of the standard MINERVA 2 would thus correspond to the connections within such a network, with the strengths of those connections being rapidly modifiable via Hebbian learning. Unfortunately, most of the essential details of how the model might actually be implemented were not provided. For example, nothing was said about how the echo intensity or content would be calculated in such a network.

In J. Culbertson, A. Perfors, H. Rabagliati & V. Ramenzoni (Eds.), *Proceedings of the 44th Annual Conference of the Cognitive Science Society.*

Ans et al. (1998) provided one possible network implementation in their *multiple-trace memory model* of word identification. This model borrows many of the core assumptions of MINERVA 2 to provide a computationally explicit account of how multisyllabic French words can be learned and then later retrieved as a function of a variety of variables (e.g., word frequency and spelling-to-pronunciation consistency) by beginning readers, skilled readers, and dyslexic readers. In this model, lexical knowledge is represented within a network of nodes, with episodic-memory nodes providing a "hub" to link orthographic input nodes, orthographic output nodes, and phonological output nodes. (The episodic-memory nodes are also connected to a pair of "reading mode" nodes that allow the model to encode and retrieve whole words, syllables, or graphemes.) Although the multiple-trace memory model is certainly successful in its own right, the complexity of the model's many other assumptions (which are specifically related to word identification and not basic memory phenomena) obscure the relationship between it and MINERVA 2, making it difficult to ascertain if and how the standard version of the latter might be implemented within a more "plain vanilla" connectionist network.

More recently, Kelly et al. (2017) provided an extensive theoretical analysis of MINERVA 2 and its relationship to other memory models (e.g., *CHARM*: Eich, 1985; *Matrix model*: Humphreys et al., 1989; *TODAM*: Murdock, 1995). This analysis indicates a formal equivalence among the models, with MINERVA 2 being mathematically equivalent to a fourth-order tensor (i.e., 4-dimensional matrix). Kelly et al. then go on to make claims about the neural implementation of MINERVA 2 that, although undoubtedly accurate, like Hintzman (1990), leave many critical details unspecified. For example, Kelly et al. claim that the similarity between a probe and trace can simply be computed as the dot product of the two vectors, and that biologically plausible neural networks are capable of performing such computations (e.g., Eliasmith, 2013). This unfortunately requires one to imagine what such an implementation would actually look like—an endeavor that is too reliant on (usually faulty) human reasoning. Kelly et al. also make one questionable claim: that a neural implementation of MINERVA 2 would be problematic or implausible on the grounds that the core assumption of the model—that individual experiences are represented by discrete memory traces—would be tantamount to positing the existence of "grandmother cells" responsible for representing individual concepts, experiences, objects, etc. In their words:

> "Modelers using MINERVA are generally agnostic as to how the model is related to the brain. No one claims that for each new experience the brain grows a new neuron that is forever singly dedicated to that particular experience. But no other interpretation of how MINERVA can be implemented in neurons has been previously proposed, leaving open the question of MINERVA's neural plausibility" (Kelly et al., 2017, p. 143).[1]

[1] Kelly et al. (2017) use "MINERVA" to refer to the entire class of models based on the original MINERVA 2 model.

In rejecting this notion, Kelly et al. describe a variant of MINERVA 2 in which individual memory traces are replaced by holographic vectors in which information is distributed across the entire fourth-order tensor of the model. Although this provides a viable interpretation of how the representations and algorithms of MINERVA 2 might be implemented in underlying neural structures, it is important to note that the notion of localist representations (i.e., "grandmother cells") has not been completely discredited, and that, to the contrary, there is considerable evidence supporting their existence (e.g., see Bowers, 2009). It also stands to reason that the average human brain contains vast numbers of un- or underutilized neurons that could be recruited singly or collectively to represent new experiences. Bearing this hypothesis in mind, the goal of this article is to show exactly how such a model might be implemented. In other words, our goal is to show how MINERVA 2, as it is standardly implemented and conceptualized, might be implemented within a connectionist network that requires only the simplest of assumptions. Before doing this, however, we first review the standard version of MINERVA 2 (see Hintzman, 1984, 1986, 1987, 1988).

Standard MINERVA 2 Implementation

As Figure 1 shows, MINERVA 2 makes a basic distinction between primary and secondary memory. Primary memory consists of a single *N*-dimensional vector representing the current contents of consciousness, while secondary memory consists of a (in principle, vast) collection of *M* such vectors. Each element of the vector is a feature that represents a perceptual or cognitive "primitive" (e.g., color, texture, animacy, emotional tone etc.) and that collectively represent the contents of our experiences.

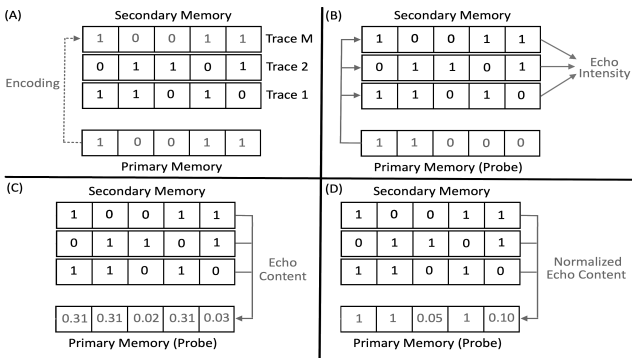

Figure 1. A schematic diagram of MINERVA 2.

In most simulations, the values of the features are determined randomly (e.g., being set equal to 1, 0, or -1 with equal probability; Dougherty et al., 1999; Hintzman, 1987, 1988; Reichle & Perfetti, 2003) or sampled from a Gaussian distribution (Kelly et al., 2017). In other simulations, however, the features are selected non-randomly to have a certain

similarity structure among the traces (e.g., high vs. low-level distortions of category prototypes; Hintzman, 1986) or to systematically represent specific types of features (e.g., letters and/or phonemes of words; Ans et al., 1998; Kwantes & Mewhort, 1999; Reichle, 2021).

As Figure 1A shows, during any interval of time, the contents of primary memory can be encoded into secondary memory as a new memory trace. This happens probabilistically, with each active feature in primary memory having a probability, equal to the value of the parameter L, of being copied into a new trace in secondary memory. Although the contents of secondary memory are fairly stable, individual features can also be forgotten (i.e., their values revert back to 0), with the probability of this happening during a given interval of time being defined by a second parameter, F. According to this conceptualization of learning and forgetting, a value of 0 indicates the absence of a particular feature in primary or secondary memory, which in the case of secondary memory can arise either because the feature was not encoded or was encoded but then forgotten. As shown in Figure 1A, many simulations for convenience simply use $L = 1$ and $F = 0$ so that a veridical copy of the contents of primary memory is simply encoded into secondary memory as a memory trace.

As shown in Figure 1B-D, a set of features in primary memory can be used as a probe to activate traces in secondary memory and thereby generate an echo having both intensity and content. To do this, the similarity between the probe and each trace i, S_i, is first calculated using Equation 1. In this equation, P_j is feature j of the probe, $T_{i,j}$ is feature j of trace i, N is the total number of features, and N_R is the number of non-zero features in either the probe or the trace. Alternatively, N_R can be set equal to the number of features, N. These two methods of calculating S_i are identical in situations where all of the probe or trace features have non-zero values. But in situations involving missing features (i.e., values of 0), the second method of calculating similarity will result in smaller values of S_i because N will typically be larger than N_R. The second method may also be problematic in situations where the number of features is correlated with the information being represented in memory because a probe containing many features will (on average) be more similar to the traces than a probe containing fewer features. For example, if the features are used to represent letters, then the number of letter features will, by definition, increase with word length, causing probes containing longer words to be more similar to their traces than probes containing shorter words. Thus, if traces can represent words up to ten letters in length, then a probe containing a 3-letter word (e.g., *cat*) can only have a maximal similarity of $S_i = 0.3$, whereas a probe containing a 7-letter word (e.g., *leopard*) can have a maximal similarity of $S_i = 0.7$. Irrespective of how similarity is calculated, the values of S_i will vary between -1 and 1, with values close to -1/1 indicating a high degree of dissimilarity/similarity between a probe and trace, and values close to 0 indicating independence (i.e., orthogonality) between a probe and trace. (S_i thus behaves like a correlation coefficient, r.)

$$S_i = \sum_{j=1}^{N} P_j T_{i,j} / N_R \quad (1)$$

After the similarity between the probe and each trace has been determined, each trace's activation is calculated using Equation 2, where the activation of trace i, A_i, is simply equal to its cubed similarity. Although Hintzman (1986, Footnote 5) notes that other values of the exponent are possible, using an odd-valued exponent preserves the sign (positive vs. negative) of the similarity. Larger values also increase the signal-to-noise ratio by selectively increasing/decreasing the activation of traces that are highly similarly/dissimilar to the probe. For example, a probe that is only modestly similar to a trace (e.g., $S_i = 0.4$) will engender only minimal trace activation ($A_i = 0.06$), whereas a probe that is twice as similar to the same trace (e.g., $S_i = 0.8$) will cause the trace to become more than eight times as active ($A_i = 0.51$).

$$A_i = S_i^3 \quad (2)$$

Finally, all of the activated traces are then used to generate a response from memory, called an *echo*. This echo has two qualities. The first is a scalar value called the *echo intensity*, or I, that reflects the sum of the activation across the M traces in secondary memory, as described by Equation 3. When used in conjunction with an appropriate decision threshold, the values of I in response to probes can be used to make "old" versus "new" recognition judgements or judgments about recency or frequency (Hintzman, 1988).

$$I = \sum_{i=1}^{M} A_i \quad (3)$$

The second quality of the echo is its *content*. The echo content consists of a composite pattern of features in which the value of each feature is weighted by each trace's activation. More formally, as Equation 4 shows, the value of each feature in the echo content, C_j, is equal to the sum across the M traces of the product of each trace's activation, A_i, and the value of its feature j, $T_{i,j}$. Because information about multiple items can be stored in a single memory trace, the features of one item can be used as a probe to recall the features of the other(s). For example, to simulate paired-associate learning, two words (A and B) can be encoded within a single memory trace, allowing the features of one word (A) to be used as probe to generate an echo content containing the features of the other (B). This pattern-completion capacity thus provides the means for MINERVA 2 to simulate a variety of findings related to recall (Hintzman, 1987, 1988). And because the echo content is a composite that is weighted by each trace's similarity to the probe, it tends to reflect the central tendency of the traces that contribute most to the echo content. For example, if a set of traces represent features that are generated from a prototype, then the echo content will be similar to that prototype even though the prototype itself may have never been encoded. MINERVA 2 thus provides an account of how concepts might be abstracted from specific instances,

allowing the model to dispense with the need for separate episodic versus semantic long-term memory (Hintzman, 1986).

$$(4) \qquad C_j = \sum_{i=1}^{M} A_i \, T_{i,j}$$

Because the echo content is a composite that reflects the global contents of memory, it will typically be "noisy," with the values of individual features being spurious. Although several memory models assume that the patterns that are recalled from memory are simply "cleaned up" using some type of separate pattern-recognition system (e.g., see Eich, 1985; Humphreys et al., 1989; Murdock, 1995), MINERVA 2 provides two ways of removing the noise from the echo content. The first involves using the echo content as a probe to generate another echo content (i.e., using Equations 1, 2, and 4). This second echo content will more closely resemble the memory trace that the original (first) probe most resembled. By reiteratively probing with the echo content, this resemblance will continue to increase until the final echo content closely resembles a trace in secondary memory (within some margin of error). The number of probe iterations required to do this can also be used to simulate the time required to retrieve information from memory, although one limitation of this approach is that the number of iterations is typically small (e.g., fewer than ten), allowing for only coarsely graded predictions about response latencies.

The second way in which MINERVA 2 removes noise from the echo content is via normalization. As Equation 5 shows, this entails identifying the feature *j* having the largest absolute value and then dividing all of the *N* features by this value. This operation normalizes the feature domain to the range [-1, 1], thereby magnifying even small differences among the values of the original echo content features.

$$(5) \qquad N_j = C_j / \max\left[\left|C_{j \in N}\right|\right]$$

As indicated previously, the standard implementation of MINERVA 2 describes how memory functions in terms of representations and algorithms (Marr, 1982), with the representations being memory traces (i.e., clusters of perceptual and semantic features) and the algorithms being those that generate the echo intensity and (normalized) content. Although there are principled arguments for why models of cognition at this level should or must be developed and evaluated independently of any consideration of their underlying neural implementation (e.g., see Coltheart, 2012), we maintain that it is none-the-less informative to consider how the representations and algorithms that support some cognitive capacity *might* be implemented. It is in this spirit of exploration that we now describe one such possible implementation of the MINERVA 2 model.

Connectionist Implementation of MINERVA 2

Figure 2 shows one possible connectionist implementation of MINERVA 2. This particular implementation was chosen because it is simple and requires only assumptions that are commonly used with connectionist networks. Consistent with what Hintzman (1990) indicated, our network implementation of MINERVA 2 consists of a single layer of connections between two layers of nodes: the first representing the features of both the input (probe) and output (echo content), and the second representing individual instances or experiences.

Hintzman (1990) suggests that the weights between the feature and instance nodes can be rapidly learned using Hebbian learning, where the change in the weight from feature node *i* and instance node *j*, $\Delta w_{j,i}$, is given by:

$$(6) \qquad \Delta w_{j,i} = \varepsilon \; act_j \; act_i$$

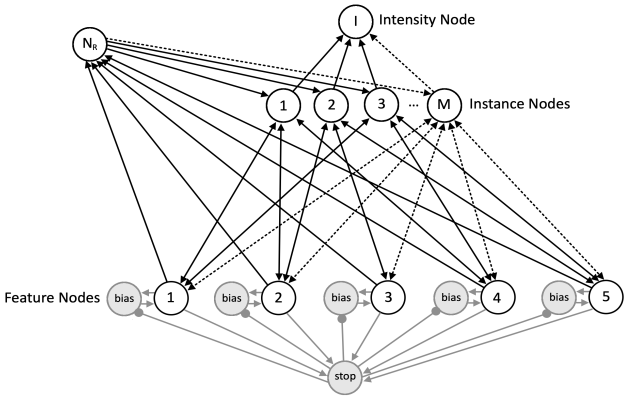

Figure 2. A connectionist implementation of MINERVA 2.

In Equation 6, ε is a constant of proportionality, or learning rate parameter, and act_j and act_i are the activation values of nodes *j* and *i*, respectively. By this account, each new experience recruits an instance node (or group of nodes) from a vast pool of undedicated nodes. The selected instance node is then coactivated with the nodes representing the features, allowing the Hebbian learning algorithm to rapidly adjust the weights (using $\varepsilon = 1$) so that, if the pattern of features recurs at some later time, it will reliably activate its corresponding instance node(s). The connection weights between the feature and instance nodes are thus isomorphic to the abstract memory trace features in the standard implementation of MINERVA 2. A comparison of Figures 1 and 2, for example, will indicate that the features [1, 1, 0, 1, 0] in memory trace #1 correspond to the connection weights between instance node 1 and features nodes 1-5. (Note that for the sake of clarity, only the non-zero weights are shown in Figure 2; the solid arrows between feature nodes 1, 2, and 4 thus correspond to the non-zero features in memory trace #1.) More incremental or graded learning can be simulated by using smaller values of ε to allow the connection weights to take on values that would (on average) approximate the encoding of features that might be expected using a value of *L* (in the standard implementation of MINERVA 2) less than 1 and/or a value of *F* greater than 0.

But how does a pattern of active feature nodes activate an instance node, act_i? As Equation 7 shows, the net input to instance node *i* (which determines its level of activation) preserves the similarity between the pattern of active feature nodes and the patterns of features that were originally associated with each instance node. Equation 7 is thus isomorphic

to Equation 1. As indicated, the net input to instance node i, net_i, equals the sum across the N feature nodes of each feature node j's activation multiplied by the weight between it and an instance node i, $w_{j,i}$. Thus, if one compares Equations 1 and 7, P_j corresponds to act_j and $T_{i,j}$ corresponds to $w_{j,i}$. Finally, as with Equation 1, the sum of the products in Equation 7 is divided by N_R, the number of non-zero feature nodes j or non-zero connection weights to hidden node i.

$$(7) \quad net_i = \left(\sum_{j=1}^{N} act_j \, w_{j,i}\right)/N_R$$

Because N_R corresponds to the union of non-zero feature nodes j (i.e., non-zero probe features) and connection weights to hidden node i (i.e., non-zero trace features), the information about the feature nodes must somehow be made available to instance node i. This is done by summing the activation of feature nodes j via the connections (having weight = 1) to and from the node labeled N_R, thereby allowing instance node i to calculate N_R using Equation 8. As shown, this equation consists of three terms: (1) the sum of the active feature nodes j; (2) the sum of the non-zero weights (i.e., $w_{j,i}$ where $act_j = 1$) connecting to hidden node i; and (3) the sum of the non-active weights (i.e., $w_{j,i}$ where $act_j = 0$) connecting to hidden node i. The third sum is subtracted from the first two so that the intersection of the union is not counted twice. This makes the information required to calculate N_R available to hidden node i so that its activation, act_i, can then be determined. Of course, a simpler method is to set N_R equal to the number of features, N. Using this alternative scheme, N_R can either be set equal to the number of feature nodes or the total number (i.e., non-zero and zero) of connections to an instance node. This second method has limitations, however, as discussed in relation to the standard MINERVA 2.

$$(8) \quad N_R = \sum_{j=1}^{N} act_j + \sum_{i=1}^{M} w_{j,i}^{act_j=1} - \sum_{i=1}^{M} w_{j,i}^{act_j=0}$$

Because connectionist networks use a wide variety of activation functions (Williams, 1987), it is trivial to convert Equation 2 into Equation 9, where S_i and A_i in the former respectively correspond to act_i and net_i.

$$(9) \quad act_i = net_i^3$$

Finally, as indicated in the Introduction, one unanswered question concerns the precise manner in which the activated instance nodes are used to generate echo intensity and content. The former is fairly easy to generate; the activation of the M instance nodes is simply summed using Equation 10. In this equation, act_I is the activation of a node (labeled "Intensity" in Figure 2) that sums the activity of the instance nodes—a quantity that corresponds to the echo intensity—with the weights between this node and the instance nodes, $w_{i,I}$, being set equal to 1. Equation 10 is thus isomorphic to Equation 3, with the main difference being that the former provides an implementational description of how the activation is actually summed across the instances. (Remember that the standard implementation of MINERVA 2 assumes that the trace activation is summed but does not describe how this actually happens.)

$$(10) \quad act_I = \sum_{i=1}^{M} act_i \, w_{i,I}$$

Turning now to the echo content, its calculation entails propagating the instance-node activation back to the feature nodes to generate a composite pattern of features that reflects each instance node's activation. This is done using Equation 11, where act_j is the activation of the feature node j, act_i is the activation of instance node i, and $w_{j,i}$ is the connection weight between the two.

$$(11) \quad act_j = \sum_{i=1}^{M} act_i \, w_{j,i}$$

Finally, we ended our discussion of the standard implementation of MINERVA 2 by describing the two methods for "deblurring" a noisy echo content. The first method of reprobing with the echo content to reduce the noise across successive iterations of echoes can be employed with the connectionist implementation of MINERVA using Equations 7, 8, 9, and 11. The second method of normalizing the echo content requires a few additional assumptions, as illustrated in Figure 2. These assumptions are related to *bias* terms or nodes (indicated by the gray nodes labeled "bias") and a special node (labeled "stop") that halts the operation of the bias nodes. Bias nodes can be conceptualized as nodes that provide additional sources of activation, biasing the activation level of another node. By this conceptualization, $bias_j$ provides an additional source of activation that incrementally increases the activation of node j so that it attains its final normalized value. How does this happen?

When the echo content is first generated, the activation of feature node j (i.e., act_j) is copied to its corresponding bias node j via the connection weights between the two nodes ($w_{j,bias} = 1$). Then, with each (arbitrary) time step, some proportion of the bias node activation, as determined by the parameter τ, is calculated (using Equation 12) and then added to act_j (using Equation 13). (Equation 13 thus adds one term to Equation 11 to reflect the activation being propagated by the bias node.) But as the activation of the feature nodes continues to ramp up over time, the feature nodes propagate their activation to the node labeled "stop." If the activation of any one feature node exceeds its minimum/maximum (i.e., $|act_j| \geq 1$), then the "stop" unit immediately inhibits the bias nodes, thereby halting any further incrementing of the feature nodes. When this happens, the final activation values of the feature node will equal their normalized values (i.e., the same values that would be obtained using Equation 5 of the standard MINERVA 2 implementation.)

$$(12) \quad bias_j = \tau \, act_j \, w_{j,bias}$$

$$(13) \quad act_j = \left(\sum_{i=1}^{M} act_i \, w_{j,i}\right) + bias_j$$

Because the number of time steps needed for the feature nodes to become normalized as described above reflects the

overall quality of the original echo content, this method of calculating the normalized echo content provides a novel way to simulate response latencies—one in which the features in the echo content "settle" into a stable pattern over time[2].

Simulation Results

To verify the equivalency of the two new versions of MINERVA 2 (i.e., the standard model using the new method for normalizing echo content and the connectionist implementation), we replicated a simulation reported by Hintzman (1986) to examine how category size (i.e., 3, 6, or 9 exemplars per category), the number of forgetting cycles (0 vs. 1), and the type of the retrieval probe (i.e., an old, previously encoded exemplar, category prototype, and low- and high-distortion versions of the prototype) affected categorization accuracy. (For methodological details of the simulation, see Hintzman, 1986, pp. 415-418.) The simulation results using 10,000 statistical subjects are shown in Figure 3 for the standard model (panels a, c, e) and the connectionist implementation (panels b, d, f)[3].

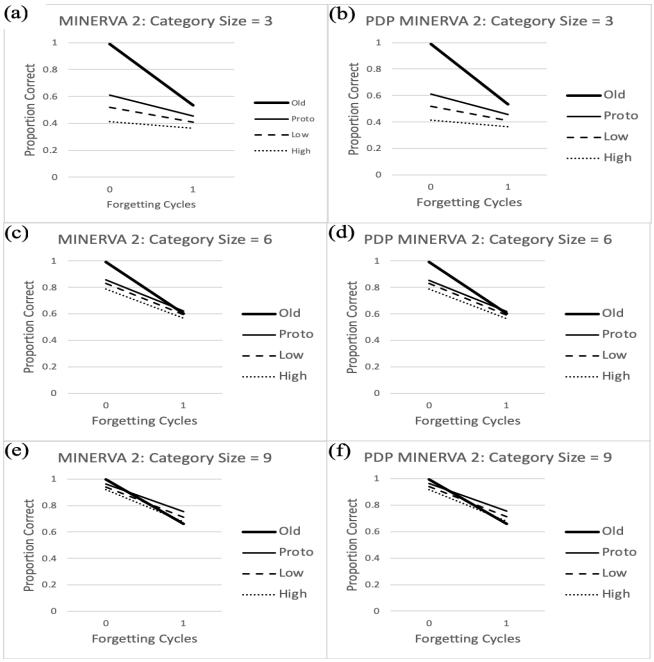

Figure 3. Simulation results.

The performance of the two models was nearly identical, with the largest discrepancy in accuracy (= 0.0006) due to trials in which the normalized echo content matched two category prototypes equally well, resulting in responses based on guessing. The results are also similar to those of Hintzman (1986, Figure 5); for example, with no forgetting, performance is best using an old exemplar as a probe and worst using a highly distorted version of the prototype, but with the difference being attenuated for larger categories, and with the prototype being more effective than an old exemplar after forgetting, especially with larger categories.

General Discussion

This article has described one possible implementation of the MINERVA 2 model of human memory (Hintzman, 1984). This implementation is likely only one of many that are possible (e.g., see Kelly et al., 2017), but is unique in that it was intended to be as simple as possible, using only assumptions that are widely employed in connectionist (neural network) models. The benefits of this exercise are at least twofold.

First, implementing the echo normalization process within the framework of a connectionist network resulted in a novel way of normalizing the echo content in the standard version of MINERVA 2—one that allows for finer-grained predictions about memory retrieval latencies.

The second, perhaps more significant benefit of this exercise is that it provides a concrete example supporting Marr's (1982) tripartite distinction among the task, representation/algorithm, and implementation levels. As documented earlier, the standard representation/algorithm-level version of MINERVA 2 has been used to simulate a wide variety of different cognitive tasks (cf., Ans et al., 1998; Dougherty et al., 1999), demonstrating that basic principles of the model (e.g., instance-based learning) play central roles in a variety of different behaviors. And similarly, the standard version of MINERVA 2 can be implemented in a variety of ways (cf., Ans et al., 1998; Kelly et al., 2017), demonstrating that the mapping between cognition and neural systems is not one-to-one (Coltheart, 2012).

Finally, it is important to acknowledge that there are degrees of biological realism, and that a few of the assumptions used in our connectionist implementation of MINERVA 2 (e.g., bidirectional propagation of activation) are seemingly at odds with what is currently known about neurophysiology and thus might have to be modified. Similarly, one might argue that a truly biologically plausible version of MINERVA 2 should be implemented using a spiking neural network in which the functional units simulate the action potentials of neurons (e.g., Eliasmith, 2013). Despite these limitations, however, we maintain that our new implementation of MINERVA 2 is more biologically plausible than the original, demonstrating how an instance-based theory of memory can be instantiated as a network of interconnected instance and feature nodes (see also McClelland, 1981).

[2] This method of generating the normalized echo content and predictions about the time required to do so can also be adopted with the standard MINERVA 2 implementation. To do this, the feature values of the initial echo content (C_j; see Equation 4) are incremented by the quantity (C_j * τ) across successive time steps until a feature value equals or exceeds its minimum/maximum (i.e., $|C_j| \geq 1$). At that time, all of the values of the features will equal N_j (as described by Equation 5).

[3] The Java source code for these simulations is available from the first author upon request.

Acknowledgements

This research was supported under Australian Research Council's Discovery Projects funding scheme (project number DP190100719).

References

Ans, B., Carbonnel, S., & Valdois, S. (1998). A connectionist multiple-trace memory model for polysyllabic word reading. *Psychological Review*, *105*, 678-723.

Bowers, J. S. (2009). On the biological plausibility of grandmother cells: Implications for neural network theories in psychology and neuroscience. *Psychological Review*, *116*, 220-251.

Collins, R. N., Milliken, B., & Jamieson, R. K. (2020). MINERVA-DE: An instance model of the deficient processing theory. *Journal of Memory and Language, 115*, 104151.

Coltheart, M. (2012). The cognitive level of explanation. *Australian Journal of Psychology*, *64*, 11-18.

Dougherty, M. R. P., Gettys, C. F., & Ogden, E. E. (1999). MINERVA-DM: A memory processes model for judgments of likelihood. *Psychological Review*, *106*, 180-209.

Eich, J. M. (1985). Levels of processing, encoding specificity, elaboration, and CHARM. *Psychological Review*, *92*, 1-38.

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. Oxford, UK: Oxford University Press.

Goldinger, S. D. (1998). Echoes of echoes? An episodic theory of lexical access. *Psychological Review*, *105*, 251-279.

Hintzman, D. L. (1984). MINERVA 2: A simulation model of human memory. *Behavior Research Methods, Instruments, & Computers*, *16*, 96-101.

Hintzman, D. L. (1986). "Schema abstraction" in a multiple-trace memory model. *Psychological Review*, *93*, 411-428.

Hintzman, D. L. (1987). Recognition and recall in MINERVA 2: Analysis of the "recognition failure" paradigm. In P. Morris (Ed.), *Modeling cognition: Proceedings of the international workshop on modeling cognition* (pp. 215-229). London, UK: Wiley.

Hintzman, D. L. (1988). Judgments of frequency and recognition memory in a multiple-trace memory model. *Psychological Review*, *95*, 528-551.

Hintzman, D. L. (1990). Human learning and memory: Connections and dissociations. *Annual Review of Psychology*, *41*, 109-139.

Humphreys, M. S., Bain, J. D., & Pike, R. (1989). Different ways to cue a coherent memory system: A theory for episodic, semantic, and procedural tasks. *Psychological Review*, *96*, 208-233.

Jamieson, R. K. & Mewhort, D. J. K. (2009). Applying an exemplar model to the artificial-grammar task: Inferring grammaticality from similarity. *The Quarterly Journal of Experimental Psychology*, 62, 550-575.

Kelly, M. A., Mewhort, D. J. K., & West, R. L. (2017). The memory tesseract: Mathematical equivalence between composite and separate storage memory models. *Journal of Mathematical Psychology*, *77*, 142-155.

Kwantes, P. J. & Mewhort, J. K. (1999). Modeling lexical decision and word naming as a retrieval process. *Canadian Journal of Experimental Psychology*, *53*, 306-315.

Marr, D. (1982). *Vision*. San Francisco, CA, USA: Freeman.

McClelland, J. L. (1981). Retrieving general and specific information from stored knowledge of specifics. In *Proceedings of the annual meeting of the cognitive science society*.

Murdock, B. B., Jr. (1995). Developing TODAM: Three models for serial-order information. *Memory & Cognition*, *23*, 631-645.

Reichle, E. D. (2021). *Computational models of reading: A handbook*. Oxford, UK: Oxford University Press.

Reichle, E. D. & Perfetti, C. A. (2003). Morphology in word identification: A word-experience model that accounts for morpheme frequency effects. *Scientific Studies of Reading*, *7*, 219-237.

Schacter, D. L., Eich, J. E., & Tulving, E. (1978). Richard Semon's theory of memory. *Journal of Verbal Learning and Verbal Behavior*, *17*, 721-743.

Williams, R. J. (1987). The logic of activation functions. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing, vol. 1: Foundations*. Cambridge, MA: MIT Press.