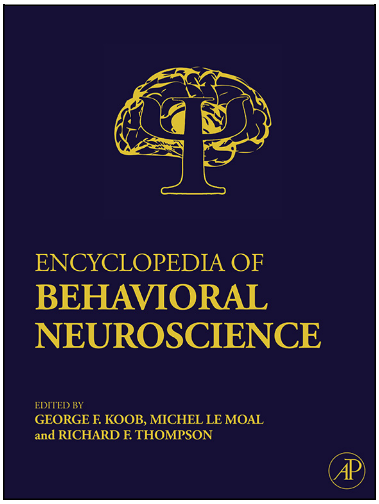

ENCYCLOPEDIA OF
BEHAVIORAL
NEUROSCIENCE

EDITED BY
GEORGE F. KOOB, MICHEL LE MOAL
and RICHARD F. THOMPSON

Sederberg P B and Norman K A (2010) Learning and Memory: Computational Models. In: Koob G.F., Le Moal M. and Thompson R.F. (eds.) *Encyclopedia of Behavioral Neuroscience*, volume 2, pp. 145–153 Oxford: Academic Press.

# Learning and Memory: Computational Models

**P B Sederberg and K A Norman**, Princeton University, Princeton, NJ, USA

## Glossary

**Episodic memory** – Memory for specific events in the past.

**Mental context** – Any information that is actively represented in a person's brain at the time they are processing a particular stimulus.

**Semantic memory** – Memory for meanings.

The goal of learning and memory research is to understand how we store and retrieve information based on our experiences. Toward this end, computational models provide formal implementations of memory theories; these formal implementations facilitate hypothesis testing and the generation of novel predictions. Computational models of memory are constrained in two directions. One goal of memory modeling research is to capture the behaviors that participants exhibit during memory tasks. Another goal is to explain how the brain gives rise to these behaviors. High-level, abstract models focus on reproducing behavior but not neural data, whereas biologically based models attempt to explain both neural and behavioral data. This article focuses on models of declarative memory, which can be divided into two components: episodic and semantic memory. The first section of the article describes abstract models of episodic memory, our ability to remember specific, previously experienced events. The second section describes abstract models of semantic memory – our ability to learn and remember the meanings of stimuli. The final section describes the Complementary Learning Systems (CLS) model, which seeks to account for both semantic and episodic memory phenomena within a single, biologically plausible computational framework.

## Abstract Models of Episodic Memory

Episodic memory experiments typically consist of a study phase (where subjects are exposed to a set of stimuli) followed by a test phase. The test phase takes the form of either a recognition-memory test (where subjects have to discriminate between studied and nonstudied stimuli) or a recall test (where subjects have to retrieve specific details from the study phase of the experiment). Abstract models of episodic memory try to describe the mental algorithms that support performance on recognition and recall tests, without specifically addressing how these algorithms might be implemented in the brain. Although there is considerable diversity within the realm of abstract episodic memory models, most of the abstract models that are currently in use share a common set of properties: Individual memories, commonly refered to as memory traces, are typically represented as vectors – where each element of that vector represents a particular feature of the memory. At study, memory traces are placed separately in a long-term store. Because of this 'separate storage' postulate, acquiring new memory traces does not affect the integrity of previously stored memory traces. At test, the model computes the match between the test cue and all of the items stored in memory. This item-by-item match information can be summed across all items to compute a 'global-match' familiarity signal. Most abstract models posit that subjects make recognition-memory judgments based on the strength of the global-match familiarity signal (i.e., the stronger the match, the more likely it is that the item was studied). Some abstract models that conform to this overall structure are Search of Associative Memory (SAM) model (first implemented by Raaijmakers and Shiffrin), Retrieving Effectively from Memory (REM) model (first implemented by Shiffrin and Steyvers), and MINERVA 2 (developed by Hintzman).

In abstract models, the same 'match' rule that is used to compute the global-match familiarity signal is also used when simulating recall tasks, although the specific way in which the match rule is used during recall differs from model to model. For example, MINERVA 2 simulates recall by computing a weighted sum of all of the items stored in memory, where each item is weighted by how well it matches the test cue. In contrast, models like SAM and REM use the individual match scores to determine which (individual) memory trace will be recalled.

Collectively, abstract models have been very successful in explaining behavioral recall and recognition data from normal subjects. They have also been used to explain data from memory-impaired subjects, by finding a set of parameter changes that lead to the desired pattern of memory deficits. The remaining part of this section presents a detailed description of the REM model. REM is highlighted because of its principled mathematical foundation, and because (of all of the models mentioned above) it is the abstract model that is being developed and applied most actively.

## The REM Model of Recognition and Recall

The REM model, first published by Shiffrin and Steyvers in 1997, is the most recent iteration of a line of models that dates back to the SAM model that was published by Raaijmakers and Shiffrin in 1981. One of the main differences between REM and previous models like SAM and MINERVA 2 is that REM implements a principled Bayesian calculation of the likelihood that the cue 'matches' (i.e., corresponds to the same item as) a particular stored memory trace, whereas the match calculation was not defined in Bayesian terms in previous models.

In REM, items are vectors of features whose values are geometrically distributed integers. The primary consequence of feature values being distributed geometrically is that high-feature values are less common than low-feature values. When an item is studied, the features of that item are copied into an episodic trace for that item via a two-step process. First, each feature is stored with a specified probability that is a parameter of the model. If a feature is stored, then a second probability determines whether that feature is stored correctly or whether it is replaced by a value sampled from the geometric distribution. A zero value means that no value is stored for the feature.

At test, the retrieval cue is compared to each stored memory trace. For each trace, the model calculates the likelihood that the cue and the trace match (i.e., they correspond to the same item). This likelihood is based on two probabilities: the probability of obtaining the observed pattern of matching and mismatching features – assuming that the cue and trace correspond to the same item – divided by the probability of obtaining the observed pattern of matching and mismatching features, assuming that the cue and trace correspond to different items.

The same core 'match' calculation is used for both recognition and cued recall in REM. The model is applied to recognition by computing the overall odds that the item is old (vs. new), calculated as the average of the likelihood values from the match calculations. If this odds value exceeds a preset criterion then the item is called 'old.' The fact that the effects of individual feature matches (and mismatches) are combined multiplicatively within individual trace comparisons and additively across traces ensures that multiple matches to a single trace have a larger effect on the odds that an item is old than the same number of feature-matches spread across multiple traces.

Recall (i.e., retrieval of specific stored details) in REM has both a sampling component (which picks a single trace out from the memory store) and a recovery component (which determines whether the sampled memory trace is retrieved successfully). The sampling probability for each item is based on the match between the memory cue and the item, scaled by the sum of the matches to all items. Once an item is sampled, the probability that the image will be recovered is based on the proportion of correctly stored item features. Thus, in REM, well-encoded items are more likely to be recovered than poorly encoded items.

Recently, Malmberg and colleagues developed a dual-process version of REM that utilizes both the 'global-match' familiarity signal and the recall process described above. When a test item is presented, the model computes the global match and it also attempts to retrieve a specific stored memory trace that matches the test item. The key benefit of using recall is that it helps the model reject lures that closely resemble studied items. For example, if subjects study 'rats' but are tested with 'rat,' this test item will trigger a strong global-match signal, but it may also trigger recall that 'rats' was studied (not 'rat'). This mismatch between the test item and the retrieved memory can be used to identify the item as a related lure. Malmberg argues that subjects primarily use this recall process to reject related lures, and that it does not play a significant role in recognizing actually studied items (this view is controversial).

### *Representative REM results*

Researchers have demonstrated that REM can explain a wide range of episodic memory findings. For example, Shiffrin and Steyvers demonstrated that the 'global-match' familiarity mechanism described above can account for the list-length, list-strength, and word-frequency effects in recognition memory.

The list-length effect refers to the finding that recognition-memory performance tends to be lower for longer versus shorter study lists. REM explains this effect because adding words to the study list increases the odds that the test item will spuriously match a stored memory trace from the study list (i.e., the model will conclude that the test item matches a stored memory trace when in fact it does not). The (null) list-strength effect refers to the finding that strengthening some list items (by repeating the items or presenting them for longer periods of time) does not impair recognition of other, nonstrengthened items. REM explains this effect because strong items have more features stored (i.e., they have more 'differentiated' memory representations) and thus are less likely to be confused with other test items. The word-frequency effect refers to the finding that words that occur with high frequency in natural language are recognized less well than low-frequency words. In REM, high-frequency words have more common feature-values than low-frequency words, which makes them more likely to be confused with other test items.

## Context and Episodic Memory

While the basic REM model provides a mechanism for how the memory system responds to a particular cue, it does not describe how the memory system behaves when external cues are less well specified, and subjects have to generate their own cues in order to target a particular memory (or set of memories). Take the scenario of trying to remember where you left your keys. The most common advice in this situation is to reinstate your mental context as a means of prompting recall – if you succeed in remembering what you were doing and what you were thinking earlier in the day, this will boost the probability of recalling where you left the keys. This idea of reinstating mental context plays a key role in theories of strategic memory search. For the purposes of this article, mental context can be defined broadly as any other information that is actively represented in a person's brain at the time they are processing a particular stimulus.

Multiple laboratory paradigms have been developed to examine strategic memory search. The most commonly used paradigm is free recall, where subjects are given a word list and are then asked to retrieve the studied word list in any order. REM can be extended to simulate free recall by adding a set of contextual features to each memory trace. For example, all of the items in the study list could be given a shared set of contextual features (effectively, a 'context tag') that signify membership in the study list. To simulate free recall, we can cue with this 'list-context' representation and sample items that were paired with the list-context representation at study. However, while this simple context-tag representation gives REM the ability to simulate free recall, it does not allow REM to simulate more nuanced features of free recall data. To fit detailed patterns of free recall data, it is necessary to specify in more detail how context changes over time, and how context is used to cue memory at retrieval.

## The Temporal Context Model

The Temporal Context Model (TCM; first published by Howard and Kahana) is the most recent in a long succession of models that use a drifting mental context to explain our ability to selectively target memories from particular time periods. The basic idea behind these models is that the subject's inner mental context (comprising the constellation of thoughts that are active at a particular moment) changes gradually over time. Early models viewed context as a vector that evolves as a function of random noise when each item is presented, with a drift-rate parameter governing the overlap of context from time-step to time-step. The main difference between TCM and previous contextual-drift models is that context does not drift randomly in TCM. Rather, contextual updating is driven by the features of the items being studied and recalled.

During the study phase of a memory experiment, two things happen when an item is presented: first, the item is associated with the state of the context vector at the time of presentation; second, context is updated by averaging together the current state of the context vector with the semantic features of the just-studied item. At test, the recall process is initiated by cuing with the current state of the context vector, which (in turn) triggers retrieval of items that were associated with these contextual elements at study. Specifically, each item is activated to the degree that the current state of context overlaps with the context that was present when that item was studied. In the most recent version of TCM, called TCM-A, these activated items compete with one another via a set of accumulators that add up evidence for each item over time (based on that item's level of activation and the activation levels of all the other items). If the level of evidence for an item reaches a prespecified threshold level, that item is recalled, and the current state of context is updated in two ways: first, by averaging in the semantic features of the just-recalled item, and second, by averaging in the state of the context vector that was present when the item was studied. This latter updating operation can be construed as 'mentally jumping back in time' to the moment when the (just-retrieved) item was studied. Once the context vector is updated, it is used to cue for additional items, which leads to additional updating of the context vector, and so on.

### *How TCM accounts for recall data*

The drifting context vector in TCM explains a number of findings in episodic recall, including both recency and contiguity effects. In TCM, the current state of context acts as the cue for memory retrieval via context-to-item associations. Because context changes gradually, the state of context at the time of test will overlap most strongly with the contexts associated with recent items. This gives rise to the recency effect seen in all episodic memory tasks. TCM can also explain the temporal contiguity effect: the finding that, when a subject recalls a particular item from the study list, they show an increased probability of (subsequently) recalling items from nearby time points in the list. TCM shows this effect because recalling an item (at test) also triggers recall of the contextual state that was present when the item was studied. This retrieved context will closely match contextual states associated with temporally proximal items, thereby making it easier to retrieve these items. For example, the contextual state associated with the fourth item on the study list will closely match the contextual states associated with the third and fifth items; thus, recalling the 'fourth-item' context will make it easier to access the third and fifth items.

## Abstract Models of Semantic Memory

Earlier, we defined semantic memory as the ability to learn and remember the meanings of stimuli. More concretely, semantic memory is our ability to construct an internal representation of the world that allows us to make predictions about 'unseen' aspects of stimuli. For example, the semantic memory system allows us to categorize the pig we see at the petting zoo as a mammal and to generalize that it has a brain and gives birth to its children alive following a gestation period, without having to take a magnetic resonance image of the pig's head or watch it give birth. The semantic memory system also allows our friends to retrieve these basic features of a pig when we recount seeing a pig at the petting zoo.

### Rumelhart Model of Semantic Cognition

We focus on the model of semantic memory developed by David Rumelhart and colleagues, because it is the simplest extant model that explains how we develop internal representations and make predictions using these representations. Our discussion here draws heavily on recent work using this model by Rogers and McClelland.

The goal of the Rumelhart model is to activate the proper set of attributes when probed with an item (e.g., 'pig') and relation (e.g., 'can'). The basic structure of the Rumelhart model (see schematic in Figure 1) consists of multiple layers of units connected in a feedforward fashion. The item units specify the objects that are being observed, the relation units specify the contexts in which we observe these objects, and the attribute units specify the 'unseen' aspects of the objects that we are trying to predict. When the network is probed by activating an item unit and a relation unit, activation spreads forward in the network (through the representational layer and the hidden layer) until it reaches the attribute layer. The spread of activation is governed by the strengths of connections between units, and the resulting pattern of activity in the attribute layer constitutes the model's prediction (e.g., 'pig' + 'can' should yield 'walk' in the attribute layer). A critical aspect of the model is that it does not prespecify what patterns should appear in the representational and hidden layers. Rather, the network learns to generate patterns in these layers (by adjusting weights) that help it predict the correct attributes. The pattern of activity in the representational layer serves as the primary representation of the item's meaning, whereas the pattern of activity in the hidden layer represents the meaning of the item in the context of a particular relation.

A central aspect of the Rumelhart model is that it learns to make better predictions by adjusting connection strengths. Learning takes place incrementally over many presentations of item, relational, and attributational patterns (i.e., seeing a fish swim in water would be represented by activating the 'fish' item unit, the 'can' relation unit, and the 'swim' attribute unit). On each trial, after generating its prediction, the network receives feedback on which attributes are actually observed for that item/relation combination. Learning in the network is driven by prediction error, that is, the discrepancy between attributes that are predicted to be present, and the attributes that are actually present. On each trial, after prediction error is computed, and weights are changed

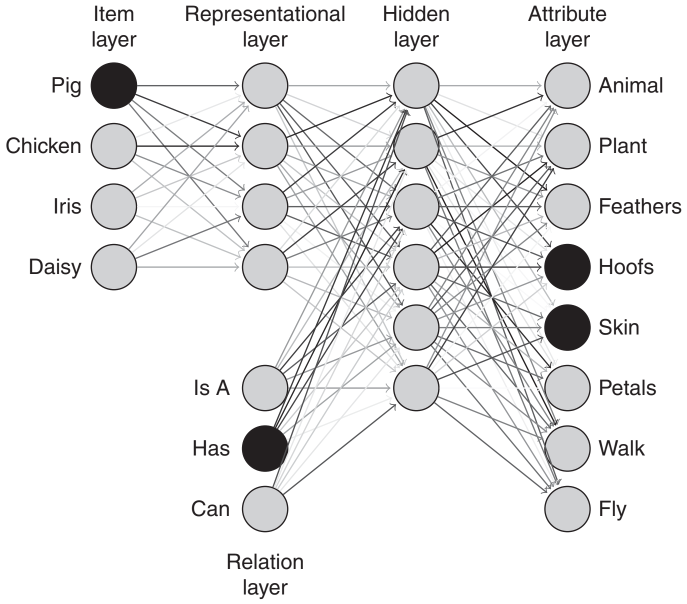

**Figure 1** Simplified diagram of the Rumelhart model.

throughout the network in order to reduce prediction error.

Specifically, Rumelhart used the backpropagation neural network learning algorithm to adjust network weights. For each unit in the network (except for item and relation units), backpropagation computes a delta ($\delta$)-value for that unit that indicates whether that unit was too active ($+\delta$) or not active enough ($-\delta$). First, backpropagation computes $\delta$-values for attribute units; then it computes $\delta$-values for each preceding layer (in turn) by multiplying $\delta$-values by network weights. For example, if an attribute unit has a positive $\delta$-value (i.e., it is too strongly active), active hidden units with positive connections to the overly active attributational unit are also assigned positive $\delta$-values (indicating that they are 'at fault' for the prediction error, and that prediction error can be reduced by reducing the activity of hidden units). After $\delta$-values have been assigned to all of the attribute-layer, hidden-layer, and representation-layer units, backpropagation changes network weights based on these $\delta$-values: if a unit is too active, weights coming into that unit (from active sending units) are reduced, and if a unit is not active enough, weights coming into that unit (from active sending units) are increased.

### *Successes of the Rumelhart model*

After a sufficient degree of training, Rumelhart showed that the model learns context-sensitive mappings between items and attributes. For example, after training, activating the 'pig' item and the 'has' relation will activate the 'skin' and 'hoofs' attributes. If instead we activate the 'pig' and the 'can' attribute, the 'walk' attribute will activate (and the 'fly' attribute will certainly not activate). This context sensitivity arises because the relation units modify the activation in the hidden layer, which is responsible for combining the activity in the representational and relational layers before activating the attribute units.

The most interesting aspect of the Rumelhart model is how internal representations (i.e., the patterns of activity in the representation and hidden layers elicited by different items) change during learning. Items with similar attributes come to elicit similar patterns of activity in the representation and hidden layers, and items with distinct attributes come to elicit distinct patterns of activity in the representational and hidden layers. For example, when training the sample Rumelhart model in Figure 1, the representations for 'pig' and 'chicken' start to converge, and these representations diverge together from the representations for 'iris' and 'daisy' because (for any given relation) 'pig' and 'chicken' are more likely to share attributes than, say, 'pig' and 'daisy.' In this way, the model learns just as a child would – by first forming a coarse representation of the environment that is refined over time based on experience.

A key property of the model is its ability to generalize to new stimuli based on their similarity to previously encountered stimuli. For example, after learning the various attributes of pigs and chickens, the Rumelhart network will be able to predict basic properties of a new animal, such as a cheetah, just by learning that it is an animal. This is because training the network to predict animal given cheetah will push the cheetah's internal representation closer to the representations of other items that predict animal (e.g., pig and chicken). This overlap in internal representations will lead to cheetah predicting other attributes that were associated with pigs and chickens (e.g., that cheetahs have skin and can walk).

## Temporal Context and Semantic Relationships

The Rumelhart model modifies its internal representations based on explicit instruction concerning which attributes should be active in a given context. Recently, several researchers have argued that meaning representations can also be acquired without explicit instruction, if the model keeps track of temporal context (i.e., it learns which items tend to be presented close in time to a given item). The key idea here is that items with similar meanings tend to occur in similar temporal contexts (e.g., couch and sofa both tend to occur close in time to other words like rug, lamp, and cushion). Given this premise, it should be possible to learn that couch and sofa have similar meanings by learning about which other words tend to co-occur with couch and sofa.

The Latent Semantic Analysis (LSA) algorithm developed by Landauer and colleagues provides a large-scale proof of the relationship between meaning and temporal context. Landauer and colleagues took a massive corpus of English texts and computed how often each word co-occurred in the same paragraph as every other word (normalized to account for differences in overall word frequency). The net product is a matrix of size $N \times N$, where $N$ is the number of distinct words in the text corpus. One way of thinking about this matrix is that each word in the matrix is represented by a 'temporal context vector' of length $N$, listing how often that word occurred with every other word. In principle, it should be possible to estimate the similarity of word meanings by looking at the similarity of the $N$-dimensional temporal context vectors associated with each word. However, Landauer also had the insight that there is considerable redundancy in the $N \times N$ co-occurrence matrix, and that (because of this redundancy) words could be represented by vectors with many fewer-than-$N$ elements. To eliminate this redundancy, LSA applies a technique called singular value decomposition (SVD) to the $N \times N$ matrix. SVD returns a set of $N$ orthogonal temporal context vectors (each of size $N$), ranked by how much variance they explain (across words) in the original matrix (technically, these are the eigenvectors of the original matrix). Landauer found that the first 300 or so of these vectors accounted for almost all of the variance in the original matrix. As such, he discarded the remaining vectors, and re-expressed each word's temporal context vector in terms of a weighted combination of these 300 'basis vectors.' The net result of this process to go from an *N*-dimensional representation for each word to a (much more manageable) 300-dimensional representation. Using these 300-dimensional vectors, Landauer and colleagues found that the cosine distances between these vectors map quite well onto the similarity values that people assign to pairs of words. For example, the cat and dog vectors are quite similar to each other (i.e., their cosine distance is very small), whereas the chicken and daisy vectors are not.

Importantly, while LSA substantiates the idea that temporal context provides information about stimulus meaning, it does not provide a mechanistic account of how the brain exploits temporal context to acquire semantic representations. Recently, Howard and colleagues argued that the TCM (described earlier) can meet these desiderata. The previous section discussed how TCM can account for episodic memory phenomena (by rapidly binding items with coactive contextual features, such that items can trigger recall of associated contexts and vice versa). TCM can be extended to address semantic learning by supplementing this rapid binding process with another learning process that gradually (across trials) learns which contextual features tend to be associated with a given item.

Another important point is that, while temporal context provides some information about word meanings, LSA-like algorithms are not meant to be a substitute for the kinds of error-driven learning that are built into the Rumelhart model. Meanings learned by LSA do not always coincide perfectly with human semantic judgments (e.g., LSA tends to give antonyms very similar meaning representations because they occur in similar contexts). To remedy these misconceptions, models of semantic memory need to receive feedback on how well they are predicting the item's attributes, and they need to be able to learn based on prediction errors. One way to update the Rumelhart model to take advantage of both error-driven learning and temporal context information would be (1) to include a representation of temporal context (akin to the representation generated by TCM) and (2) to train the model to predict the current state of the temporal context representation (plus other relevant attributes) when an item is presented. Forcing the model to predict temporal context will bias the model to assign similar internal representations to items with similar temporal contexts (just as forcing the model to predict attributes like 'animal' causes the model to assign similar representations to all of the 'animal' items).

Finally, while the above discussion focused on temporal context, other kinds of context also provide information about stimulus meaning. For example, knowing that two items tend to appear at similar spatial locations provides some information about their meanings, irrespective of whether they appear close in time to one another. The general principle of training models to predict contextual information (be it temporal, spatial, or some other type of information) will allow models of semantic memory to leverage all of these regularities when learning about meanings.

## Learning, Memory, and the Brain

The previous sections focused on abstract models of episodic and semantic memory. This section describes the CLS model, which intertwines episodic and semantic memory into a single, neurally plausible computational framework.

### The Complementary Learning Systems (CLS) Model

The CLS model (first outlined by McClelland, McNaughton, and O'Reilly) incorporates several widely held ideas about the division of labor between hippocampus and neocortex that have been developed over many years by many different researchers. (It is important to note that the CLS model is only one of many biologically-plausible models of the hippocampal role in learning and memory. For example, see Further reading for other influential hippocampal models by Gluck, Rolls, Hasselmo, and Eichenbaum.) According to the CLS model, the neocortex forms the substrate of our internal model of the structure of the environment. In contrast, the hippocampus is specialized for rapidly and automatically memorizing patterns of cortical activity, so they can be recalled later (based on partial cues). This characterization makes it clear that neocortex is the key substrate for semantic memory and that hippocampus is crucial for episodic memory, although (as discussed below) both structures contribute to both kinds of memory.

The model posits that the neocortex learns incrementally; each training trial results in relatively small adaptive changes in synaptic weights. These small changes allow the cortex to adjust its internal model of the environment gradually in response to new information. The other key property of neocortex (according to the model) is that it assigns similar representations to similar stimuli. Use of overlapping representations allows cortex to represent the shared structure of events, and therefore makes it possible for cortex to generalize to novel stimuli based on their similarity to previously experienced stimuli. In contrast, the model posits that hippocampus assigns distinct, pattern-separated representations to stimuli, irrespective of their similarity. This property allows the hippocampus to memorize arbitrary patterns of cortical activity associated with particular events rapidly without suffering from unacceptably high (catastrophic) levels of interference.

### *Norman and O'Reilly model of episodic memory*

Norman and O'Reilly constructed hippocampal and cortical networks that instantiate the CLS principles outlined above, and applied these networks to simulating episodic memory data. In both the hippocampal and cortical networks, to-be-memorized items are represented by patterns of excitatory activity that are distributed across multiple units (simulated neurons) in the network. Excitatory activity spreads from unit to unit via positive-valued synaptic weights. The overall level of excitatory activity in the network is controlled by a feedback-inhibition mechanism that samples the amount of excitatory activity in a particular subregion of the model, and sends back a proportional amount of inhibition.

The architecture of the model (illustrated in Figure 2) reflects a broad consensus concerning key anatomical and physiological characteristics of different hippocampal and cortical subregions, and how these subregions contribute to the overall goal of memorizing cortical patterns. The entorhinal cortex (EC) contains a compressed representation of information represented elsewhere in cortex. The hippocampal network memorizes patterns of EC activity by linking these patterns to a set of units (an 'episodic representation') in region CA3, which is then linked back to EC via region CA1. When a pattern is presented, connections are strengthened between active EC and CA3 units, between active units within CA3, and between active CA3 and CA1 units; collectively, these synaptic modifications allow the network to recall entire stored EC patterns based on partial cues (pattern completion). To minimize interference, the network has a built-in bias to assign relatively nonoverlapping (pattern separated) CA3 representations to different episodes. Pattern separation occurs because of strong feedback inhibition in CA3, which leads to sparse representations (i.e., representations with relatively few neurons active). The dentate gyrus (DG) assists in the pattern separation process by forming an even more sparse representation of the EC pattern, which then projects into region CA3.

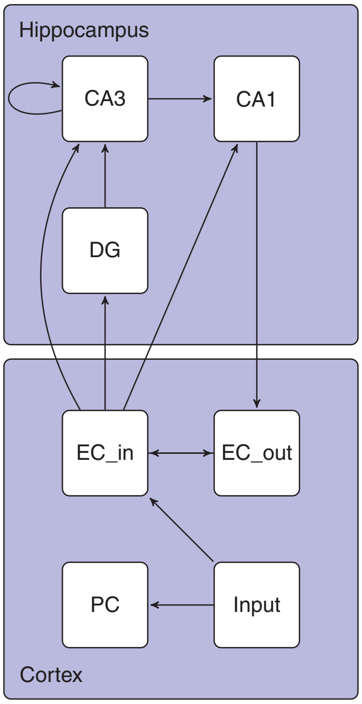

**Figure 2** Architecture of the Norman and O'Reilly Complementary Learning Systems model of episodic memory.

The cortical component of the CLS model consists of an input layer (corresponding to lower regions of the cortical hierarchy) which projects in a feedforward fashion to a hidden layer (corresponding to regions further up in the hierarchy, including perirhinal cortex (PC) and EC). As mentioned earlier, the main function of cortex is to extract statistical regularities in the environment. The two-layer CLS cortical network (where hidden units compete to encode regularities that are present in the input layer) is meant to capture this idea in the simplest possible fashion. In the Norman and O'Reilly simulations, learning in both the cortical and hippocampal subregions of the model was implemented by means of a simple Hebbian learning rule that strengthens connections between active sending and receiving neurons and weakens connections between active receiving neurons and inactive sending neurons.

As described by Norman and O'Reilly, the hippocampal and cortical networks constitute a biologically based dual-process model of recognition memory. As with the dual-process REM model described earlier, the CLS model posits that familiarity (i.e., global match) and recall of specific details both contribute to recognition memory. In the CLS model, hippocampus supports recall of specific studied details, but (because of its tendency to assign distinct CA3 representations to stimuli, regardless of their similarity) it is not well suited for computing the global match of the test cue to studied items. The cortex, on the other hand, does not learn quickly enough to support recall of details from specific events, but it can compute a scalar familiarity signal that tracks how well the test cue matches studied items. As items are presented repeatedly, their representations in the hidden layer of the cortical network become sharper: novel stimuli weakly activate a large number of hidden units, whereas previously presented stimuli strongly activate a relatively small number of units. Sharpening occurs in the cortical model because Hebbian learning specifically tunes some hidden units to represent the stimulus, and these units suppress the activity of other units (via the feedback-inhibition mechanism). Furthermore, since the cortex assigns similar hidden representations to similar stimuli, these sharpening effects generalize smoothly to other stimuli based on their similarity to the original stimulus (so sharpening tracks global match).

Norman and O'Reilly showed how, taken together, the hippocampal network and cortical network can explain a wide range of behavioral findings from recognition and recall list-learning experiments. For example, the model can account for the null-recognition list-strength effect described earlier, and it also makes the prediction that list-strength effects should be observed when recognition memory is driven by recall of specific details as opposed to familiarity. Furthermore, because the CLS model maps clearly onto the brain, it is possible to use the model to address neuroscientific data in addition to (purely) behavioral data. For example, the model correctly predicts that focal hippocampal lesions should differentially impair recognition performance on tests where distractors are very similar to studied items (because these tests benefit from the hippocampus' ability to assign distinct representations to similar patterns). The model also successfully predicts that lesioned patients' deficit on these tests can be ameliorated by giving subjects a forced choice between studied items and corresponding related lures (because the forced-choice procedure allows subjects to leverage small but reliable differences in cortical familiarity between studied items and corresponding lures).

The original form of the CLS episodic memory model also has some serious flaws. In particular, Bogacz and Brown showed that the cortical network's capacity for familiarity discrimination (i.e., the number of studied patterns that it can discriminate from nonstudied patterns with 99% accuracy) falls far below the documented capacity of human recognition memory. This problem can be traced back to the Hebbian learning rule, which is insufficiently judicious in how it adjusts synaptic strengths: it strengthens synapses between co-active units even if the memory is already strong enough to support recall, and it weakens synapses between active receiving units and other inactive units, even if those other units are not interfering with recall of the sought-after memory. This excess synaptic modification greatly increases the extent to which new learning interferes with stored knowledge. The solution to this problem is to switch to an error-driven learning rule that compares top-down expectations (generated by cortex's internal representation of the environment) to sensory inputs, and only modifies synapses when the model's expectations are incorrect. The backpropagation rule described earlier fits this description, but this rule is widely believed to be biologically implausible. Determining biologically plausible methods of enacting error-driven learning in cortex has been a major focus of computational modeling research, and researchers have devised a wide range of potential solutions to this problem (see, e.g., the work of Carpenter and Grossberg). Recently, Norman and colleagues swapped out the Hebbian learning rule for a more judicious, biologically plausible rule that uses neural oscillations to probe for 'weak points' in cortical representations; this new learning rule greatly improves the cortical model's familiarity discrimination capacity.

### *Toward a full episodic/semantic CLS model*

The Norman and O'Reilly CLS simulations focused on episodic memory. However, the CLS cortical model (equipped with an error-driven learning rule) should be able to account for all of the semantic learning phenomena that were discussed in the Rumelhart model section above; the key prerequisites for simulating these results are a learning rule that is driven by prediction error, and the ability to re-represent inputs in order to minimize prediction error. CLS can also be used to explore how hippocampo-cortical interactions shape semantic memory. One of the key claims made in the original formulation of CLS was that hippocampus could play back significant, once-presented events to the cortex during sleep, thereby allowing the slow-learning cortical network to absorb these events into its semantic network. Recent modeling work by Norman, Newman, and Perotte explored other aspects of learning during sleep (e.g., the possibility that learning during REM sleep could help to repair cortical memories that are damaged by new learning).

Another feature missing from most CLS models is a representation of context. Some CLS models have included a simple context layer (akin to the one used in the Rumelhart model), but none of these models have seriously explored how temporal context is represented in the brain. It seems likely that prefrontal cortex (PFC) will play a key role in temporal context memory (by virtue of its ability to actively maintain patterns of neural firing over time). Other researchers have noted that the EC also has some intrinsic ability to maintain information over time. Future work using the CLS framework will explore the contributions of both PFC and EC to representing temporal context.

## Key Challenges

Over the past several decades, a consensus has emerged among computational modelers regarding certain key aspects of memory functioning (e.g., how the hippocampus supports episodic memory). However, there is still extensive work to be done in specifying extant computational models of learning and memory. For example, while there is widespread agreement that cortical learning is driven by a comparison of top-down expectations versus bottom-up inputs, there is still extensive debate concerning precisely how the brain implements this learning process. In addition, extant models have focused on specifying basic encoding and retrieval processes, and have not yet addressed fundamental issues with regard to strategic influences on memory (e.g., how much to rely on recall of specific details vs. familiarity when making recognition decisions; how to strategically construct retrieval cues during memory search). This latter issue might benefit from a normative approach (i.e., mathematically deriving how subjects should be cuing memory and making decisions in order to maximize performance, and then exploring whether subjects actually use these 'optimal' strategies).

Importantly, the models of episodic and semantic memory described above constitute only a small portion of the total space of memory models. Other models have been developed to account for data from other kinds of memory tasks, such as working memory tasks (which ask subjects to actively maintain stimulus information in the face of distraction), conditioning tasks, spatial learning tasks, and motor-learning tasks. These other models leverage some of the same computational principles and neural systems described above, but they also describe important ideas that were not reviewed in the preceding sections. For example, one topic that has received extensive attention in recent years is how the brain leverages simple reinforcement signals (rewards and punishments) to improve behavior. Another topic that has received extensive attention is the control of working memory (i.e., how does the brain learn when to 'gate' information into working memory and when to release it from working memory; see the work of Frank, O'Reilly, and colleagues).

The key challenge, moving forward, will be to integrate insights gleaned from all of these models while still keeping model complexity within manageable limits. Models like CLS are quite complex as they stand, and the extensions proposed above (e.g., adding a PFC network to CLS to help it maintain temporal context) will make the models even more complex. The saving grace here is that modern-day memory models can be used to address an enormous range of findings – using a complex model to explain a single result may not be all that meaningful, but simultaneously accounting for behavioral and neural data from multiple types of learning experiments is still a worthy challenge. So long as modelers continue to apply all available constraints to theory development, we should continue to see steady progress toward a complete computational account of learning and memory data.

*See also*: Declarative Memory; Episodic and Autobiographical Memory: Psychological and Neural Aspects; Learning and Memory: Computational Models; Memory and Aging, Neural Basis of; Memory Consolidation.

## Further Reading

Bogacz R and Brown MW (2003) Comparison of computational models of familiarity discrimination in the perirhinal cortex. *Hippocampus* 13: 494–524.

Carpenter GA and Grossberg S (1988) The ART of adaptive pattern recognition by a self-organizing neural network. *Computer* 21(3): 77–88.

Gluck M, Meeter M, and Myers C (2003) Computational models of the hippocampal region: Linking incremental learning and episodic memory. *Trends in Cognitive Sciences* 7(6): 269–276.

Hasselmo M and Eichenbaum H (2005) Hippocampal mechanisms for the context-dependent retrieval of episodes. *Neural Networks* 18(9): 1172–1190.

Hazy TE, Frank MJ, and O'Reilly RC (2007) Towards an executive without a homunculus: Computational models of the prefrontal cortex/basal ganglia system. *Philosophical Transactions of the Royal Society B* 362: 1601–1613.

Hintzman DL (1988) Judgments of frequency and recognition memory in a multiple-trace memory model. *Psychological Review* 95: 528–551.

Howard MW and Kahana MJ (2002) A distributed representation of temporal context. *Journal of Mathematical Psychology* 46: 269–299.

Landauer TK and Dumais ST (1997) Solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* 104: 211–240.

Malmberg KJ (2008) Recognition memory: A review of the critical findings and an integrated theory for relating them. *Cognitive Psychology* 57: 335–384.

McClelland JL, McNaughton BL, and O'Reilly RC (1995) Why there are complementary 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.

Norman KA and O'Reilly RC (2003) Modeling hippocampal and neocortical contributions to recognition memory: A complementary learning systems approach. *Psychological Review* 110; 611–646.

Norman KA, Newman EL, and Perotte AJ (2005) Methods for reducing interference in the complementary learning systems model: Oscillating inhibition and autonomous memory rehearsal. *Neural Networks* 18: 1212–1228.

Norman K, Newman E, and Detre G (2007) A neural network model of retrieval-induced forgetting. *Psychological Review* 114(4): 887–953.

Polyn SM and Kahana MJ (2008) Memory search and the neural representation of context. *Trends in Cognitive Sciences (Regular Edition)* 12(1): 24–30.

Raaijmakers JGW and Shiffrin RM (1981) Search of associative memory. *Psychological Review* 88: 93–134.

Rogers T and McClelland J (2004) *Semantic Cognition: A Parallel Distributed Processing Approach*. Cambridge, MA: Bradford Books.

Rolls E and Kesner R (2006) A computational theory of hippocampal function, and empirical tests of the theory. *Progress in Neurobiology* 79(1): 1–48.

Sederberg PB, Howard MW, and Kahana MJ (2008) A context-based theory of recency and contiguity in free recall. *Psychological Review* 115(4): 893–912.

Shiffrin RM and Steyvers M (1997) A model for recognition memory: REM – retrieving effectively from memory. *Psychonomic Bulletin and Review* 4: 145.