Cogn Comput (2014) 6:45–65
DOI 10.1007/s12559-012-9198-4

Taxonomical Associative Memory

Diogo Rendeiro · João Sacramento · Andreas Wichert

Received: 12 January 2012 / Accepted: 22 November 2012 / Published online: 5 December 2012

Abstract Assigning categories to objects allows the mind to code experience by concepts, thus easing the burden in perceptual, storage, and reasoning processes. Moreover, maximal efficiency of cognitive resources is attained with categories that best mirror the structure of the perceived world. In this work, we will explore how taxonomies could be represented in the brain, and their application in learning and recall. In a recent work, Sacramento and Wichert (in Neural Netw 24(2):143–147, 2011) proposed a hierarchical arrangement of compressed associative networks, improving retrieval time by allowing irrelevant neurons to be pruned early. We present an extension to this model where *superordinate* concepts are encoded in these compressed networks. Memory traces are stored in an uncompressed network, and each additional network codes for a taxonomical rank. Retrieval is progressive, presenting increasingly specific *superordinate* concepts. The semantic and technical aspects of the model are investigated in two studies: wine classification and random correlated data.

Keywords Categorization · Taxonomical structure · Hierarchical clustering · Associative memory · Cell assemblies

Introduction

The significance of categorization in the cognitive sciences has been most elegantly expressed by Stevan Harnad: *"to cognize is to categorize: cognition is categorization"*[2].

D. Rendeiro (✉) · J. Sacramento · A. Wichert
INESC-ID and Instituto Superior Técnico, Technical University of Lisboa, Av. Prof. Dr. Anıbal Cavaco Silva, 2744-016 Porto Salvo, Portugal
e-mail: diogo.rendeiro@ist.utl.pt

The ability to sort out the objects and events we perceive into categories allows us to efficiently use our limited cognitive resources. Categorization simplifies mental life by reducing a plethora of objects to a manageable number of concepts (i.e., mental representations of categories). This is best achieved by categories that closely map the highly correlated structure of the environment, thus maximizing the ratio of information gain versus cognitive effort [3].

Psychologists have noted that classification systems of natural kinds are commonly used across cultures and tend to exhibit hierarchical taxonomy-like structure [4]. The quest for a better understanding of the nature of knowledge organization in the mind has also been explored in the experimental realm, namely through lesion and neuroimaging studies aiming to identify brain regions involved in processing taxonomic categories. Studies of brain-damaged patients with category-specific deficits [5, 6] credit the notion of distinct domain-specific neural correlates for distinct conceptual categories. This view finds further support in functional imaging studies [7, 8], presenting the activation of distinct neural regions for natural kinds versus artifacts. Beyond the animate/inanimate divide (which might be evolutionary motivated), further evidence has been presented for neural correlates of taxonomical organization [9], showing category-related activation of different cortical regions for a set of arbitrary categories (chairs, faces, and houses).

Inspired by these behavioral and biophysical insights, we speculate on an organizational model of cell assemblies which accounts for the coding of epistemic information.

Hierarchical Organization of Memory

Recent computational studies have shown that a hierarchical tree-like organization of associative memories can accelerate memory retrieval [1, 10]. In this model, memory

traces are distributed across a hierarchy of Willshaw associative networks [11], which are successive approximations of one another. The hierarchical arrangement allows for an early and progressive filtering of the relevant neurons, thus improving retrieval time in computer implementations or the energy requirements of fully parallel memory circuitry such as hippocampal networks.

This hierarchical setup motivated the following question: *Could these approximate versions of memory traces encode concepts at higher taxonomical levels?* In pursuit of this intuition, we set out to investigate how could a simple yet biologically well-grounded memory model [11–14] encode taxonomical information in a hierarchical fashion.

We developed an extension to the aforementioned model where learning and retrieval are aided by a taxonomy. This taxonomy is built by a computational method that successively groups pairs of similar elements into categories. Each rank of the taxonomy directly relates to an associative network that codes the categories at that rank.

The retrieval process consists of sequentially querying each network while progressively filtering neurons. Hops from one network to another equate to traveling down a path of linked categories from the most abstract category (i.e., most compressed network) to the most specific, the last layer containing accurate (uncompressed) versions of learned elements. Retrieval is progressive, presenting descriptions of *superordinate* concepts for the recalled element at every taxonomical rank.

This new taxonomical associative memory is studied in two settings. First, we apply it to the domain of wine classification in order to test its semantic capability. Additionally, random datasets are employed to gather empirical insight into technical aspects, namely computational trade-offs regarding retrieval speedup, error reduction, and storage capacity.

Concepts and Categories

Since Aristotle [15, pp. 6, 19, 120] philosophic tradition favors the use of explicit definitions to ascertain the meaning of words and the categorization of kinds [16], this definitional approach also known as *the classical view of concepts* [17] requires that every concept be mentally represented by a definition composed by necessary and jointly sufficient conditions for category membership. This legacy was evident in earlier psychological approaches to categorization. However, since then, multiple methods were developed (e.g., based on prototypes, exemplars, or rules).

Human categorization can be applied to different kinds of objects and studied in diverse situations (e.g., shape is important for visual categorization tasks) and is influenced by context (e.g., the object "apple" may be categorized as "fruit" or as "computer company"). The means by which we categorize differ by kind [18]. Therefore, it is important to choose which kinds and tasks to focus. In cultures around the world, usage of classification systems for live organisms (i.e., fauna and flora) and man-made objects is common; moreover, these two domains (also known as natural and artifact kinds) are frequent in cognitive studies of categorization. Our work shall focus the verbal categorization task regarding natural and artificial kinds.

To categorize something is to consider it belongs to a group of things. The act of thinking of an object as an instance of a class is a convenient trick, for one needs not to know about every wine in existence to have some idea of what is a red or a white wine. In essence, the mind works with concepts, mental representations of categories, instead of dealing with the details of every object. For instance, if one has for dinner a ribeye steak with a glass of Cabernet, and finds such combination pleasant, one may simply record such event as another case of *"red wine goes well with meat"*.

Similarity

The downfall of the classical view of categorization came about from both empirical problems and theoretical arguments (for a detailed review see [16]). Following Eleanor Rosch's influential studies [19] during the 1970s, the scholarly consensus regarding categorization followed a paradigm of featural representation and similarity assessments. Unlike the clear-cut Aristotelian categories, similarity-based category systems allow for a degree of fuzziness which better fits human reasoning.

Category systems tend to maximize similarity among elements of a category and minimize similarity among elements of contrasting categories [3]. To allow the matching of objects and categories, a mental representation of both is necessary. For categories, concepts can be formed to represent them by combining those features most common among category members. Categorization can thus be expressed as an assessment of how similar the mental representation of an item is to a concept [20].

Considering that objects can be described by a set of discrete features [21, 22], similarity among two objects is defined as an increasing function of their shared features and a decreasing function of their distinct features [23]. Tversky's contrast model [20, 22] is a featural measure of similarity that satisfies the aforementioned conditions. According to this model, the similarity between items a and b is a function S of their respective feature-sets A and B given by

$$S(A,B) = \gamma \cdot \phi(A \cap B) - \alpha \cdot \phi(A - B) - \beta \cdot \phi(B - A), \quad (1)$$

where the minus symbol denotes the set difference operation. The term $A \cap B$ designates the shared features of both items, while $A - B$ and $B - A$ represent the distinct features of items a and b, respectively. Function $\phi(\mathscr{S})$ measures the salience of each component in a feature-set $\mathscr{S}$, producing a weighted sum of these features. A theory of salience is outside the scope of the contrast model. Nonetheless, note that such theory should account for the intensity of features, as well as their ability to differentiate among relevant objects [20].

The contribution of each term is given by the weight factors expressed by γ, α and β. Tuning these affects how sensitive the model is to shared versus distinct features. These factors also convey properties of the model, like the asymmetry of similarity ratings, that is, with equally salient features, $S(A, B) < S(B, A)$ as long as $\alpha > \beta \wedge |A| > |B|$.

Tversky also proposed a ratio model where similarity is normalized so that $S \in [0, 1]$,

$$S(A,B) = \frac{\phi(A \cap B)}{\phi(A \cap B) + \alpha \cdot \phi(A - B) + \beta \cdot \phi(B - A)}, \quad (2)$$

with α, $\beta \geq 0$. The ratio model generalizes various set theoretical approaches proposed in earlier literature on similarity. Variations can be specified by tuning the parameters (e.g., α, $\beta = 1$; α, $\beta = 1/2$; $\alpha = 1 \wedge \beta = 0$).

Next, we will look into how the contrast and ratio models can be applied to the task of discerning an ordering of typicality among category members.

Typicality

An item is judged as typical if it is representative of its category. For instance, while a penguin is a bird, a robin is a more typical bird. Typical items tend to possess features which are common to other category members, while atypical items tend to share features with contrast categories [24]. Typicality ratings predict how easily people perform in a verbal categorization task: people take longer to categorize atypical elements than typical ones [25].

These effects can be accounted for as a measure of item-concept similarity using a simplified version of the contrast model. Let features be equally salient (i.e., $\phi(X)$ simply counts elements of a feature-set X), common and distinctive features weight equally (i.e., γ, $\alpha = 1$) and features distinct to the item are disregarded ($\beta = 0$). The similarity of feature-set B (representing item b) to a concept $\mathscr{C}$ is given by

$$S(\mathscr{C},B) = |\mathscr{C} \cap B| - |\mathscr{C} - B|. \quad (3)$$

For different items b_1 and b_2 belonging to category $\mathscr{C}$, b_1 is more typical than b_2 as long as $|\mathscr{C} \cap B_1| > |\mathscr{C} \cap B_2|$, that is, b_1 shares more prototypical features. The difference in categorizing typical and atypical items in terms of speed can be related to how the similarity assessment is computed. Assuming a given threshold of similarity must be attained to categorize an item, and considering that cognitive effort (i.e., *time* × *resources*) is spent in evaluating matching and distinctive features, a higher ratio of matching features to distinctive features represents a higher probability of early termination.

While Eq. 3 accounts for differing ratings of typicality among members of a category, it does not apply for the task of deciding to which category an object belongs. For such a decision, contrasting categories must be factored in. A sensible approach [26] is to select the highest similarity rating of an item to various concepts. To achieve this, the feature-set counts should be normalized by the number of prototypal features of category $\mathscr{C}$ expressed as $|\mathscr{C}|$,

$$S(\mathscr{C},B) = \frac{|\mathscr{C} \cap B|}{|\mathscr{C}|} - \frac{|\mathscr{C} - B|}{|\mathscr{C}|} = \frac{2}{|\mathscr{C}|} \cdot |\mathscr{C} \cap B| - 1 \quad \in [-1, 1]. \quad (4)$$

A more generic metric that was suggested for this task consists of a ratio between the similarity of an item to a target category and its similarity to all alternative categories. [20, 27]

Alternatives to Similarity

While elements of featural categories are similar by their shared properties, members of a relational category share a common relational structure (e.g., for X to be a bridge, X must connect two other entities) [28].

Categorization approaches based on rules (also referred to as relations or theories) are more resource intensive than their featural counterpart. Learning of relational categories requires more attributes to be processed than learning featural categories [29]. While relational reasoning is flexible and powerful, it lacks the cognitive efficiency of the simpler featural approach. Neuropsychological and neuroimaging studies [30] present evidence for the existence of both featural and rule-based categorization systems based on distinct memory systems. Rule-based categorization tasks activate brain areas associated with a working memory system, whereas tasks requiring similarity assessments between an item and a concept (via prototypes and exemplars) are associated with an explicit long-term memory system.

Recent work [31] comparing featural and relational categorization suggests these processes may actually work together. By encoding useful relations as new basic features, common relations could be readily identified without stressing cognitive resources [32]. Conversely, these new

features could serve as input to the discovery of new relations capable of differentiating among objects.

Taxonomical Structures

Across cultures, classification systems of natural kinds have been found to possess hierarchical structure [4]. A taxonomy is a hierarchical classification system, composed as a tree-like arrangement of categories, which are related to one another by means of class inclusion. Categories higher in the hierarchy are said to be *superordinate* relative to those beneath them, whereas those lower in the hierarchy are said to be *subordinate* relative to those above them. With the exception of the highest level category, each category within a taxonomy is included in a superordinate category [33].

Hierarchical category systems are said to possess a vertical and an horizontal dimension [3]. Along a vertical dimension, categories vary according to their level of inclusiveness, from the most abstract (e.g., animal) to the most specific (e.g., cf. Fig. 1, animal → mammal → dog → bulldog). At a given level of abstraction, various contrasting categories are present (e.g., cf. Fig. 1, beneath mammal, dog vs. horse).

The utility of a hierarchical description lies in the nature of the class inclusion relation [16]. Class inclusion is *asymmetric* (e.g., not all animals are necessarily dogs), *transitive* (e.g., if all dogs are mammals and all mammals are animals, then all dogs are animals) and supports *properties inheritance* (e.g., if all animals breathe and all dogs are animals, then all dogs breathe).

In essence, taxonomies maintain knowledge about related entities, and make it accessible via *transitivity* and *feature inheritance*. Given a taxonomy, new properties learned about a category can be generalized to categories at lower levels of abstraction [16].

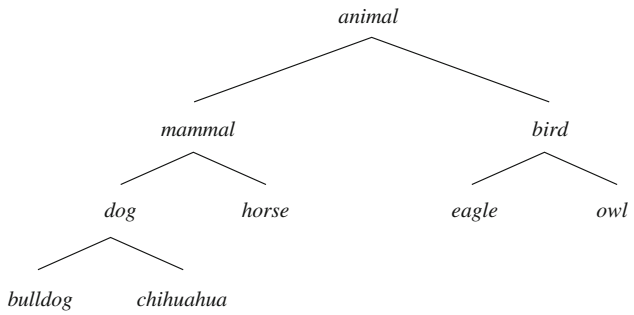


Fig. 1 A simplified conceptual tree-like hierarchy. *Lines* represent generalization/specialization relationships linking concepts to superordinates and subordinates, respectively

Differentiated Categories

Within a taxonomy, concepts at varying levels of abstraction present different properties. Between *superordinate* and *subordinate* categories, a middle level of specificity called the *basic level of categorization* is the preferred level at which categorization occurs (e.g., we prefer the category *dog* rather than more specific or general alternatives such as *bulldog* or *animal* respectively). Basic-level categories are most differentiated [24], which is to say they are balanced in terms of informativeness versus distinctiveness.

Highly informative concepts also have a higher degree of within-category similarity (i.e., more common features), whereas highly distinctive concepts have higher between-category dissimilarity (i.e., more distinct features).

Subordinate categories, found at deeper levels of a taxonomy, are informative (common features outnumber distinct features) but not very distinctive. On the other hand, superordinate categories, found at higher taxonomical levels, are distinctive but not informative (distinct features outnumber common features). Basic concepts score highly on both accounts. However, in addition to being informative, atypical subordinates are also distinctive (e.g., penguins are not very similar to other birds) and, in certain categorization tasks, preferred to their basic-level counterparts [34].

Numerical Taxonomy

In biological systematics, numerical taxonomy concerns the ordering of related taxa into taxonomical ranks through numerical methods, that is, the application of computers to taxonomy. Within numerical taxonomy, phenetics (or taximetrics) deals with the classification of organisms on the basis of their overall similarity, evaluated by their measurable traits [35]. These attributes are encoded as binary variables (i.e., present or absent) and weighted equally. In spite of this, complex attributes can be broken down into simpler attributes, each one counting toward similarity evaluations as a single unit of information. Thus, the salience of a complex feature- –which can denote intensity and/or ability to differentiate among objects—can be encoded by multiple (sub-)features.

Sneath and Sokal [35] describe a two-step numerical method for producing a taxonomy. First, for n taxa, $n(n-1)/2$ similarity coefficients are computed, comparing every taxon with every other. These coefficients are organized in a $n \times n$ lower triangular matrix cf. Fig. 2, with fixed (maximal) value in the principal diagonal (i.e., every taxon is completely similar to itself). Second, through cluster analysis, taxons are re-organized into taxa ranks according to their proximity, as depicted in Fig. 3.

In this setting, similarity between taxon is used as a metric, which by definition must be symmetric. Tversky's

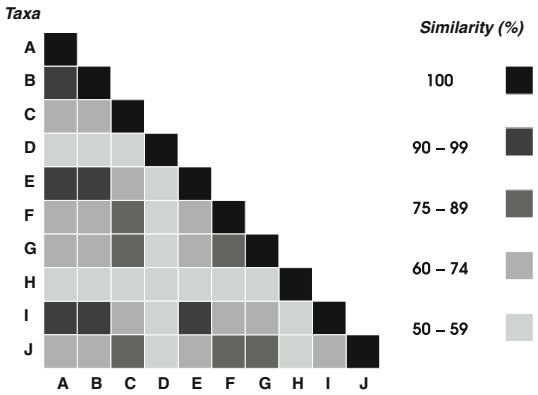


Fig. 2 Illustration of a matrix of similarity coefficients between pairs of taxa. Degree of similarity varies with *greyscale shading*, according to legend [35, 36]

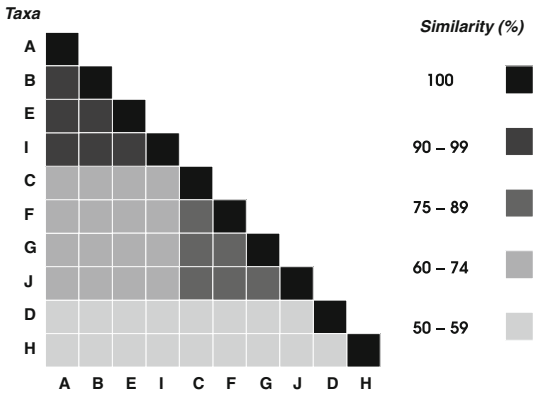


Fig. 3 Re-ordering of the coefficients in Fig. 2 by grouping similar taxa. *Triangles* are formed for high similarity values [35, 36]

contrast ratio introduced in Eq. 2 can be tuned for symmetry with $\alpha = \beta$. An appropriate measure of taxon proximity is the Jaccard index [37], an instance of the contrast ratio with γ, α, $\beta = 1$,

$$S(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad \in [0,1], \tag{5}$$

where shared and distinct features are weighted equally.

The technical details of the clustering methods are thoroughly examined in cluster analysis literature, see e.g., [38–40]. The basic hierarchical clustering procedure starts with all taxa as individual clusters and successively merges the two closest (i.e., most similar) clusters until only one remains. Hierarchical clustering methods differ in their definition of inter-cluster distance, and consequently, in their outputs as well. The most common hierarchical clustering methods applied in phenetics, as highlighted by Sokal in a later work [41], are single linkage, average linkage, and complete linkage.

Let pairwise distance among taxa $d(x, y)$ be defined as $1 - S(x,y)$. Given two clusters $\mathscr{A}$ and $\mathscr{B}$, single linkage clustering defines the distance D between them by their most similar (closest) taxa,

$$\min\{d(x,y) : x \in \mathscr{A}, x \in \mathscr{B}\}, \tag{6}$$

whereas in complete linkage clustering, D is defined by their most dissimilar (distant) taxa,

$$\max\{d(x,y) : x \in \mathscr{A}, x \in \mathscr{B}\}. \tag{7}$$

Single link, with its local merge criterion, tends to group distant taxa, linking them by a series of relatively close intermediate taxa, a phenomenon known as *chaining*, whereas complete linkage on the other hand is susceptible to outliers.

The average linkage method expressed in equation 8 factors all pairwise similarities among taxa in the assessment of inter-cluster distance—defined as the average of distances of all pair of taxa from different clusters—thus avoiding the pitfalls of the other methods, which equate cluster distance with a single pairwise similarity [40].

$$D(\mathscr{A},\mathscr{B}) = \frac{1}{|\mathscr{A}| \cdot |\mathscr{B}|} \sum_{x \in \mathscr{A}} \sum_{y \in \mathscr{B}} d(x,y), \tag{8}$$

Mental Representations

We will now explore how classification systems are mentally represented, and how they can be implemented by the circuitry of the brain.

Hierarchical structure is an universal aspect of classification systems. Thus, our inquiry regarding the cognitive representation of classification systems begets the question of how a hierarchy of concepts could be represented in the mind.

One theory that is consistent with connectionist models [42–44] calls for a tree-like memory structure where a set of connections represents a hierarchical network of linked concepts and their attributes [16]. This pre-stored hierarchy model is supported by evidence gathered in cognitive studies on transitivity and property inheritance, where subjects performed faster on cognitive tasks requiring the traversal of a small number of taxonomical links [45].

Further investigation into how to store taxonomical information requires an overview of fundamental neural models of memory that describe the dynamics of learning and recall.

Neural Associative Memories

In a biological assembly of neural cells (or neurons), information is stored at the *synapses*, the contact points between neurons. Processing is done by signals neurons receive and may react to. The basic anatomy of a neuron includes *dendrites*, the input transmission channels where signals from other neurons are received at the *synapses*; the

axon, the channel for transmitting output signals, and the *soma* (cell body) where simple computations based on input signals occur that might trigger the neuron to fire. These four elements constitute the minimal structure required to artificially model a neuron [46].

The pioneering artificial neural associative memory was the *Lernmatrix* due to [47]. The *Lernmatrix* was object of extensive study since its inception, see, for example, [11–13, 48–52].

Independently of Steinbuch's *Lernmatrix*, in 1969 David Willshaw published *Non-holographic Memory* [11], describing a similar model and providing the basis for the first formal studies of associative memory models.

A rigorous analytical treatment of the associative memory model was provided by Günther Palm [12], who defined neural associative memories as a family of artificial neural networks which implement the functions of classification and memory by way of a mapping F between *discrete* input X and output spaces Y. In other words, an associative memory stores a finite set of M associations

$$S := \{(\mathbf{x}^{\mu} \mapsto \mathbf{y}^{\mu}) : \mu = 1, \ldots, M\}, \tag{9}$$

between pairs of patterns which are represented by binary vectors $\mathbf{x} \in \{0,1\}^m$ and $\mathbf{y} \in \{0,1\}^n$.

Representing Synapses

An artificial neural network with an input layer of m neurons connected to an output layer with n units is typically modeled as an adjacency matrix of n columns and m rows, where alike a graph theory representation, non-zero elements represent connections between elements.

These models use weights to represent the strength of synaptic connections. The state of a network is described by a weight matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$

$$\mathbf{W} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1n} \\ W_{21} & W_{22} & \cdots & W_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{m1} & W_{m2} & \cdots & W_{mn} \end{pmatrix} \tag{10}$$

where each column represents the weight vector of each processing unit. In this work, we assume weight matrices are always initialized with all synapses in the inactive state, that is, with all elements $W_{ij} = 0$.

Lernmatrix

Hebbian Learning

Canadian neuropsychologist Hebb [53] put forward a greatly influential theory on how learning occurs. His idea that synaptic connectivity is reinforced in response to coincident pre- and post-synaptic activity becomes so pervasive in the connectionist community that its advocates summarized it as a slogan: *"what fires together, wires together"*.

The *Lernmatrix* stores a *clipped* binary synaptic weights matrix $\mathbf{W} \in \{0,1\}^{m \times n}$ obtained by a *clipped* version of Hebbian learning expressed as

$$W_{ij} = \min(1, W_{ij} + x_i y_j). \tag{11}$$

Let us denote the activity level of a pattern by the number of its active elements, that is, $|\mathbf{x}| := \sum_i x_i$. It has been shown [11] that the patterns stored according to the clipped rule (11) must be *sparse*, that is, $|\mathbf{x}^{\mu}| \ll m$ and $|\mathbf{y}^{\mu}| \ll n$. The intuition is to keep the connectivity matrix $\mathbf{W}$ from being full of "ones" before storing a large number of patterns M; for a rigorous information-theoretic treatment see [12, 52].

Threshold Strategy

Associative retrieval is initiated by presenting a cue pattern $\tilde{\mathbf{x}}$ to the network input. Each output[1] neuron j will calculate its *dendritic potential* by performing a weighted sum over the active inputs. Afterward, the network applies a transfer function to translate dendritic potential sums into binary valued components of the output vector $\hat{\mathbf{y}}$,

$$\hat{y}_j = \begin{cases} 1 & \text{if } \sum_{i=1}^{m} W_{ij} x_i \geq \Theta, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Quality of retrieval depends on the careful choice of the global threshold Θ. Too low a value will result in *add-errors*, while high values shall produce *miss-errors*. If $\tilde{\mathbf{x}}$ is incomplete but not noisy, the optimal solution is $\Theta = |\tilde{\mathbf{x}}|$ [11].

This dynamic threshold control models a uniform, global feed-forward inhibitory field that could be implemented by a partner interneuron, balancing the activity level of each neural unit as a function of the total input activity.

Other more complex threshold-inhibition strategies are required in the presence of complications such as input noise or synapse failure [51, 54, 55].

Tree-like Associative Memory

In an effort to reduce the computational footprint of retrieval of Willshaw-type memories in sequential implementations, as well as their metabolic counterpart in the brain, a novel structural representation for the associative memory task [1] was proposed, drawing inspiration from the Subspace Tree indexing method [56].

[1] Note that in auto-associative setups, every neuron performs input and output roles.

In a serial implementation of the associative memory task, a recall operation is sequentially performed by every neuron. By exploiting the sparseness of the input pattern with an index vector representation, it is possible to avoid unnecessary comparisons at each neuron. The sparseness of stored patterns also implies that for any given query, only a few neurons will actually fire. Thus, if a computationally inexpensive filter function could be used to test whether a neuron will fire for a given input pattern, retrieval operations could be significantly reduced. It is highly desirable of such a function that it does not produce false negatives, for these could lead to missing one-entries.

Interestingly, such a filter function can be achieved by a smaller associative memory storing compressed versions of the input patterns. By recursion, a hierarchy of successively compressed memories of decreasing resolution could be employed, progressively pruning neurons during retrieval. This hierarchical ensemble of R memories is formally described as an ordered set of R synaptic matrices

$$\mathscr{W} = (\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(r)}, \ldots, \mathbf{W}^{(R)}), \tag{13}$$

where each matrix $\mathbf{W}^{(r)}$ has fixed address space dimension m and variable number of neurons n_r. The R-th associative memory has $m \times n$ dimension and stores the associations between the original uncompressed patterns. A guarantee that no false negatives occur in the compressed patterns is the single requirement posed on the compression technique. This restriction can be satisfied by a Boolean OR aggregation over input patterns. In short, the compression method generates compact content patterns by computing Boolean OR's over non-overlapping windows of size a_r.

During learning, for each association $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ stored at the full uncompressed memory, an association with a compressed content pattern $(\mathbf{x}^\mu, \zeta_r(\mathbf{y}^\mu))$ $\forall r : 1 \leq r < R$ is also stored at every compressed r-th memory. ζ_r is defined as a family of functions $\zeta_r : \{0,1\}^{n_{r+1}} \rightarrow \{0,1\}^{n_r}$ which recursively approximates content patterns. This recurrence is defined component-wise from one level to the next

$$[\zeta_r(\mathbf{y}^\mu)]_i = \bigvee_{j=i\cdot a_r-(a_r-1)}^{i\cdot a_r} [\zeta_{r+1}(\mathbf{y}^\mu)]_j \tag{14}$$

The content-space dimensions of the memories, $n_1 < \ldots n_r < \ldots < n_R = n$, are in inverse proportion to the aggregation window factors, $a_1, a_2, \ldots, a_R = 1$, and are expressed recursively as

$$n_r = \begin{cases} \lceil n_{r+1}/a_r \rceil & \text{if } 1 \leq r < R, \\ n & \text{if } r = R. \end{cases} \tag{15}$$

The structure of the hierarchy of memories is governed by the number R of memories and their respective aggregation window factors a_r. The optimal hierarchy parameterization [10] that maximizes retrieval efficiency for a Willshaw net is given by $a_r^{opt} \in \{2, 3\}$ and $R^{opt} \leq \log(n/l)$ where the activity level l is sparse (i.e., $l \sim \log(n)$).

Associations are learned at the r-th memory by a clipped Hebbian rule

$$W_{ij}^{(r)} = \min\left(1, \sum_{\mu=1}^{M} x_i^\mu \, [\zeta_r(\mathbf{y}^\mu)]_j\right). \tag{16}$$

Retrieval is initiated by presenting an incomplete or distorted query pattern $\tilde{\mathbf{x}}$ to the lowest resolution memory at $r = 1$. The same recall cue is used at every other memory in the hierarchy. However, at $r + 1$, only a subset of neurons will have to perform a dendritic sum.

Any one entry in a component j of the output pattern $\hat{y}^{(r)}$ retrieved at the r-th memory corresponds to an index set $Y_j^{(r)}$ with a_r elements that identify the uncompressed units at the next level $r + 1$:

$$Y_j^{(r)} = \{j \cdot a_r, j \cdot a_r - 1, \ldots, j \cdot a_r - (a_r - 1)\}. \tag{17}$$

These $|\hat{y}^{(r)}|$ non-empty sets can be merged to form the complete set $\mathscr{Y}_{r+1}$ of indices for which the dendritic sum must be calculated at level $r + 1$:

$$\mathscr{Y}_{r+1} = \bigcup_{j:\hat{y}_j^{(r)}=1} Y_j^{(r)}. \tag{18}$$

During recall, the dendritic sum at the $r + 1$th memory of this hierarchical setup is very much similar to that of a classic Willshaw associative memory, with the important caveat of being restricted to the indices of $\mathscr{Y}_{r+1}$. The same applies to the threshold cut. Hierarchical retrieval is therefore expressed as

$$\hat{y}_j^{(r+1)} = \begin{cases} H[(\sum_i W_{ij}^{(r+1)} \tilde{x}_i) - \Theta] & \text{if } j \in \mathscr{Y}_{r+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

Taxonomical Associative Memory

The argument from natural cognition for cognitive economies arising from taxonomical organization of knowledge motivated us to computationally investigate potential retrieval benefits in a hierarchical memory model that can cope with correlated data. We study retrieval performance (measured by neural activation levels) and accuracy (measured in terms of retrieval error). Our work is based on a simple associative network that learns through Hebbian plasticity.

In our hierarchical model, the retrieval process is fully distributed and allows for bit-correction from incomplete pattern cues. Moreover, it is progressive: at each taxonomical level of our model, we extract a pattern which represents an intermediate categorization of the memory trace being

recalled. Progressively, we obtain a finer-grained resolution of the pattern (i.e., feature-set) being recalled.

A simplifying assumption of our model is a symbolic step, prior to learning, where a clustering procedure extracts the correlational structure of the input data, defining feature-set partitions for every category. A neurally plausible online implementation of the clustering procedure that also allows for robust coding and generalization is an important starting point for future work. We comment on the requirements posed for such process on "Online Learning".

Technical Description

In the hierarchical model proposed in [1], a set of compressed networks acts as a filter in the retrieval operation, pruning unnecessary neurons. A Boolean OR-based transform is used to code the approximate patterns stored in the intermediate networks, satisfying a single prerequisite posed on the compression technique: no false negatives may occur.

We propose a semantically enriched extension to this model (hereafter referred to as the prime model) where filter networks play a dual role by encoding superordinate concepts in addition to their technical purpose in the hierarchical retrieval process.

As in the prime model, our hierarchical memory is formally described as an ordered set of R associative memories with dimension $m \times n_r$, that is, with varying number of neurons $n_1, n_2, \cdots n_R$. In our model, the uncompressed associative memory at depth R is necessarily auto-associative ($n_R = m$). At every other level r, a binary 1/0 pattern $\mathbf{x}^\mu$ is stored as an association $(\mathbf{x}^\mu, \xi_d(\mathbf{x}^\mu))$, where $\xi_d(\mathbf{x}^\mu)$ is a compressed version of $\mathbf{x}^\mu$ which encodes a higher-order concept at the taxonomical level d.

In the prime model, approximate versions of a binary pattern $\mathbf{x}^\mu$ are produced by calculating a partition of $\mathbf{x}^\mu$ onto n_r binary sub-vectors of dimension m/n_r, and aggregating these sub-vectors with the Boolean OR operation. This approach disregards which features are merged together.

In contrast, in our taxonomical model, we require that each OR aggregate preserves some semantic description of $\mathbf{x}^\mu$ by encoding a superordinate concept which contains $\mathbf{x}^\mu$. To achieve this, each semantic aggregate merges only features that co-occur in the category it represents.

The memory capacity of the traditional Willshaw network is dependent on sparsity requirements which are not present in a dataset with semantically close examples where the feature distribution is not uniform. In this setting, cliques of neurons tend to merge and retrieval is prone to add-error. The retrieval prescription of the prime hierarchical model will predictably not help, as the contiguous OR-based aggregates will reproduce the problem into the filter networks. If semantic aggregates are chosen that efficiently distinguish contrasting concepts at every level, our model will be able, through efficient pruning, to reduce add-noise at the uncompressed network.

At variance with the generic hierarchical model, where its structure is defined by the number of R memories and the aggregation window factors a_r, the structure of the proposed model is given by a clustering process. In this setting, R corresponds to the number of levels of the resulting taxonomic tree, and partitions are non-sequential, variable-sized and possible overlapping, thus a_r is not applicable.

Taxonomical Properties

Consider a set $\mathscr{A}$ of M items described by binary 1/0 patterns of m features. In order to store each pattern $\mathbf{x}^\mu \in \mathscr{A}$, we must compute $R - 1$ approximate versions of every pattern for every intermediate network. To do so, we shall first elicit the taxonomical relationships within $\mathscr{A}$ by applying an appropriate hierarchical clustering technique (e.g., average linkage cf. Eq. 8) over $\mathscr{A}$ using a measure of pairwise pattern similarity (e.g., Jaccard index cf. Eq. 5). The resulting taxonomy $\mathscr{T}$, built from the step-by-step merge of cluster pairs, provides a hierarchical compositional description of $\mathscr{A}$.

A taxonomy $\mathscr{T}$ can be viewed as a binary tree with $P = 2M - 1$ nodes, where each node represents a cluster C_p, with $p = 1 \ldots P$, and each link between nodes has a height representing the taxonomical distance between the respective clusters. Out of P nodes, M leaf nodes represent single-element clusters. For simplicity, we will disregard the height of the links and focus on the structure of $\mathscr{T}$. This approach allows us to directly exploit the structure of the taxonomic tree, where if we were to preserve cluster distances, a procedure for flattening the tree would be required, adding complexity. The implicit trade-off is that we are discarding information and allowing clusters to form at the same taxonomic rank that are at different levels of abstraction (and thus with varying semantic properties such as informativeness vs. distinctiveness).

The last merged cluster C_1 is at the root node in the binary tree depiction of $\mathscr{T}$. At the next level, we find two new clusters contained in C_1. The process is repeated for each cluster, unless it is a leaf node. Leaf nodes have no child nodes and may occur at any level in the taxonomy. At the deepest level D of the taxonomy, all nodes are leaf nodes. At any given level of abstraction d, the number of clusters cannot be greater than 2^{d-1}. Each non-single-element cluster represents a category, and each taxonomical level d represents a level of abstraction.

To illustrate every step in the setup of our model, we will use a simple set of five elements, each describing a

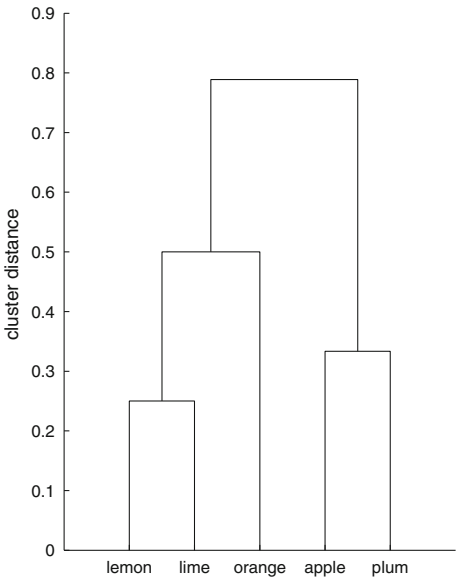


Fig. 4 Dendrogram illustrates the hierarchical clustering procedure for a simple set of fruits. Cluster distance determines the order in which these are merged

fruit, detailed in Table 1. By applying our preprocessing clustering method over the illustrative dataset, a taxonomic tree is produced as shown in Fig. 4.

As Table 2 shows, clusters higher in the concept hierarchy contain more elements which share less features.

Given the relation between taxonomic height and our memory model, the fruit dataset requires three auxiliary networks, the first of which distinguishes clusters C_2 versus C_3, the second codes for clusters C_4 versus C_5, and single-element clusters C_6 and C_7 and so on.

Representing Clusters

At the core of our taxonomical memory, model lies the ability to represent, store, and retrieve information about the clusters that compose the taxonomy. Each intermediate network will code clusters of a given taxonomical rank. In detail, we store associations between each learned pattern and its respective cluster at the taxonomical rank represented by that network.

To achieve this, we first require a code to represent each cluster at a given taxonomical level as a binary pattern which can be stored in the appropriate intermediate filter networks.

The coding must be able to identify any cluster that exists at a given taxonomical level, including leaf nodes at previous levels. It is also desirable that the codes present low activity levels, to avoid overloading of the intermediate networks that will store them.

Presently, we describe a positional coding scheme which represents clusters with binary patterns and abides by the restrictions posed above. Let c_d count the number of clusters at level d, and let k_d count the number of single-element clusters (leaf nodes) at previous levels. A binary pattern of $(c_d + k_d) \leq 2^{d-1}$ bits with one active unit can describe any cluster at level d, as well as any single-element cluster at previous levels. Let function $f_d(C_p)$ produce such binary pattern $\mathbf{c}$ (with $|\mathbf{c}| = 1$) describing C_p by its relative position within a taxonomy.

The interpretation of taxonomy $\mathscr{T}_a$ as a binary tree is shown in Fig. 5, which includes, in annotations, the coding of each cluster, illustrating the application of function $f_d(C_p)$.

By defining a taxonomical level δ for which the first intermediate memory codes, we establish a relation between the ordered set of R memories and the taxonomy $\mathscr{T}$.

During learning, each pattern $\mathbf{x}^\mu \in \mathscr{A}$ is presented to the full $m \times m$ auto-associative memory and stored as an auto-association $(\mathbf{x}^\mu, \mathbf{x}^\mu)$. At the same time, $\mathbf{x}^\mu$ is presented to $R - 1$ intermediate networks. The r-th network codes for a taxonomical level $d = r + \delta - 1$ by storing $(\mathbf{x}^\mu, \xi_d(\mathbf{x}^\mu))$ associations where the content pattern $\xi_d(\mathbf{x}^\mu) = f_d(C_p : \mathbf{x}^\mu \in C_p)$ represents the closest cluster C_p containing $\mathbf{x}^\mu$ at taxonomical level $r + \delta - 1$.

To illustrate the learning process described above, Table 3 presents item–category associations stored for the fruit dataset. The patterns representing each fruit are coded according to the ordered feature-set {sweet, sour, round,

Table 1 Featural description of fruits

Fruit	Feature-set
Apple	{sweet, hard, round}
Plum	{sweet, round}
Orange	{sweet, round, citrus, juicy}
Lemon	{sour, citrus, juicy}
Lime	{sour, round, citrus, juicy}

Table 2 Fruit clusters: shared features

Cluster	Fruits	Shared features
C_1	{apple, plum, orange, lemon, lime}	{}
C_2	{orange, lemon, lime}	{citrus, juicy}
C_3	{apple, plum}	{sweet, round}
C_4	{lemon, lime}	{sour, citrus, juicy}
C_5	{orange}	{sweet, round, citrus, juicy}
C_6	{apple}	{sweet, hard, round}
C_7	{plum}	{sweet, round}
C_8	{lemon}	{sour, citrus, juicy}
C_9	{lime}	{sour, round, citrus, juicy}

Clusters C_5 to C_9 are leaf nodes of the fruit taxonomy

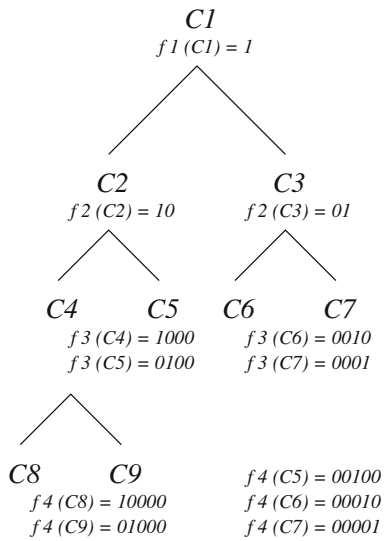


Fig. 5 A binary tree depiction of the fruit taxonomy. Annotations at each cluster show its coding $f_d(C_p)$ at a given level, including leaf node clusters at previous levels (three instances in this case: before depth $d = 4$, we find C_5, C_6, and C_7, representing *orange*, *apple,* and *plum*, respectively). Note that the notation in the image $C1, C2, \ldots Cn$ is equivalent to the one previously used $C_1, C_2, \ldots C_n$. The same applies for function $f_d(C_p)$

hard, citrus, juicy}, where ones code for feature presence (e.g., consider *plum* which is defined by the features *round* and *sweet*, its code is 101000). The clusters are represented by the aforementioned positional coding.

Furthermore, we need to map the feature-set partition represented by any given cluster, as to later map which units will be filtered during retrieval.

This is achieved by recursively computing a Boolean sum pattern using the bitwise OR operator

$$\mathscr{U}(C_p) = \bigvee_{\mu} \mathbf{x}^{\mu} \quad \forall \mu : \mathbf{x}^{\mu} \in C_p, \tag{20}$$

Retrieval of Patterns and Concepts

Our taxonomical retrieval prescription may use up to $R - 1$ filter networks, and the process can be halted at any network. At the r-th intermediate filter network, the retrieved content pattern $\xi_d(\tilde{\mathbf{x}})$ aids the pruning of relevant neurons in the next network. However, at the last traversed filter network, for every superordinate concept represented by pattern $\xi_d(\tilde{\mathbf{x}})$, we map the feature-set pattern $\mathscr{U}(C_p)$ of the respective cluster.

Afterward, we retrieve the best approximation for $\tilde{\mathbf{x}}$ at the uncompressed R-th memory, triggering only the neurons which code for the features represented in a feature-set pattern $\mathscr{U}(C_p)$.

Consider the associations learned for the fruit taxonomy depicted in Table 3. When the first auxiliary network is cued with pattern 101000 (representing *plum*), we should retrieve pattern 01, representing cluster C_3, whose feature-set $\mathscr{U}(C_3)$ is represented by pattern 101100.

Table 3 Table presenting item–category associations stored at every auxiliary network for the fruit dataset

Fruit	Lemon	Lime	Orange	Apple	Plum
$\mathbf{x}^{\mu}$	010011	011011	101011	101100	101000
1^{st}*net*	C_2	C_2	C_3	C_3	C_3
$\xi_2(\mathbf{x}^{\mu})$	10	10	01	01	01
2^{nd} *net*	C_4	C_4	C_5	C_6	C_7
$\xi_3(\mathbf{x}^{\mu})$	1000	1000	0100	0010	0001
3^{rd} *net*	C_8	C_9	C_5	C_6	C_7
$\xi_4(\mathbf{x}^{\mu})$	10000	01000	00100	00010	00001

Fruits are coded for feature presence following the order of the feature-set {sweet, sour, round, hard, citrus, juicy} Note that we consider $\delta = 2$, meaning the first auxiliary network codes for $d = 2$, distinguishing C_2 from C_3

Given this information, we may proceed with retrieval at the uncompressed 6×6 auto-associative R-th network, pruning irrelevant units from the process (which according to $\mathscr{U}(C_3)$ are {*sour*, *citrus*, *juicy*}). Using the two-unit auxiliary net, the total activated units (5 units) to perform retrieval slightly outperform direct retrieval at the uncompressed network (which requires 6 units). With larger feature spaces and sparse datasets, these benefits improve greatly.

In a prototypical fashion, the meaning of the concept represented by cluster C_p can be described in terms of shared features by

$$\mathscr{I}(C_p) = \bigwedge_{\mu} \mathbf{x}^{\mu} \quad \forall \mu : \mathbf{x}^{\mu} \in C_p, \tag{21}$$

where $\bigwedge$ is the bitwise AND operator.

In our fruit taxonomy, the shared features of each non-single-element cluster that pattern $\mathscr{I}(C_p)$ represents were previously listed in Table 2.

Shared feature patterns are important for the progressive categorization of cued patterns. In addition to the plain retrieval of patterns, our taxonomical model is capable of producing intermediate categorizations of a cued pattern $\tilde{\mathbf{x}}$ at every respective intermediate network. This is achieved by mapping the retrieved cluster-coding pattern $\xi_d(\tilde{\mathbf{x}})$ at a filter network with the appropriate prototypical feature-set described by $\mathscr{I}(C_p)$. We illustrate this with our fruit taxonomy by producing a pattern representing *lime* as a cue. A visual depiction of the path of such query in the taxonomical tree is shown in Fig. 6. The output of concept feature-sets for the cluster at every level is detailed in Table 4, along with the elements of said cluster (for reference).

The proposed cluster representations highlight the degree of within-category similarity through a measure of feature sharedness. This sort of representation allows

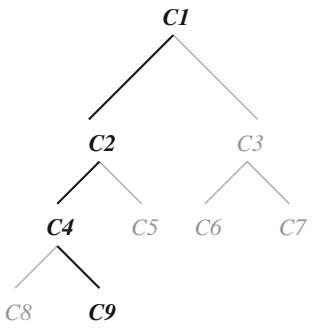


Fig. 6 Search path in a taxonomical tree for the retrieval of the pattern representing a *lime*

Table 4 Concept retrieval example: *What is a lime?* It is a fruit , more specifically a citrus (C_2), not sweet like an *orange* but sour (C_3), and unlike a *lemon* which has an oval shape, a *lime* is perfectly round (leaf node C_9)

Lv.	C_p	Fruits	Shared features ($\mathcal{I}(C_p)$)
2	C_2	{orange, lemon, lime}	{citrus, juicy}
3	C_4	{lemon, lime}	{sour, citrus, juicy}
4	C_9	{lime}	{sour, round, citrus, juicy}

inference processes: by observing an object's feature which is highly shared within a category, one might positively adjust one's estimate that said object belongs to said category.

However, to fully leverage this sort of representation, it must be exploited in the context of contrasting categories (a feature that is highly shared by elements that belong to several contrasting categories does not provide much information). A categorization system can be said to be useful to the extent that it improves one's ability to accurately predict object properties given category knowledge and vice-versa.

Multiple metrics for category goodness have been proposed [57–59], most notably the notion of cue validity [57] prominent in Rosch's studies [3] on the acquisition of the so-called basic categories. The metric we find most appropriate is that of category utility [60], which maximizes the probability of common features within a category while minimizing the probability of common features between elements of contrasting categories.

In conclusion, it is convenient to have the notion of feature sharedness embedded in category representation as it plays a significant role in categorization.

Related Hierarchical Models

Artificial neural network models such as multi-layer perceptrons [61] where learning is accomplished by the back propagation algorithm [62] or Kohonen's self-organizing maps [63] have been applied to model complex psychological phenomena in a hierarchical fashion in a range of applications including phoneme recognition [64] and the development of infant cognition [65]. These models are capable of generalization and classification when presented with real-world data with rich and complex statistics, such as inter-class correlations or highly variable activity levels.

We opted for a minimalistic approach, based on a conservative, local Hebbian learning rule. During the learning phase of the Willshaw model, coincidental firing events are encoded in a Hebbian manner via a local synaptic plasticity rule. Unlike most artificial neural network models where the synaptic strengths are analog variables, here synapses are switch-like and can only assume two stable values. The continued interest over this extremely simplified model of plasticity was initially driven by hardware implementation considerations and mathematical tractability, as well as near-optimal storage capacity for sparse random (uncorrelated) binary patterns [12, 52, 66].

Interestingly, physiological recordings in hippocampal networks of Sprague-Dawley rats suggest that CA3-CA1 synapses might operate in a digital fashion [67, 68]. This finding has further motivated theoretical research on Willshaw-type learning, especially in the context of forgetful (palimpsest) networks where synapses are bistable and led by a combination of potentiation and depression [69–73]. For simplicity, we do not include depression in our learning dynamics and assume that the number of patterns that is presented to the network is parameterized and kept finite. A natural future extension of the present work is to replace the static Willshaw rule with a dynamical process *ála* Amit and Fusi [69] so that forgetting takes place and new items lead to graceful degradation of older ones.

Our work is related to a previous extension of the canonical Hopfield model [74], where a hierarchical arrangement of layers is set up so that the system recovers high storage capacity in the presence of hierarchically correlated patterns [75]. These patterns are assumed to have an ultrametric correlational structure and are generated according to a stochastic process where the activities of patterns that belong to a given cluster are conditionally dependent on the activity of a cluster prototype pattern, as in the artificial dataset generation procedure that we present in "Random Correlated Data". In a similar fashion to our network, there is a sequentially constrained retrieval dynamics where the threshold of each subsequent neuron depends on the activity of the precedent layer. However, the employed synaptic plasticity rule is non-local and depends on the joint activity of descendants and prototypes, and there is an increase instead of a decrease in retrieval complexity: all layers have the same number of neurons, and no filtering process occurs, all of the synapses have to be checked whenever a cue pattern arrives.

Experimental Evidence

Semantic Dataset

To test our taxonomical model, we built a semantically rich exemplar dataset on the domain of wine tasting. The data were iteratively groomed through a systematic comparative review of tasting notes in social wine websites, and wine characteristics at winery and producers websites.

The resulting dataset contains descriptions of $M = 51$ wines. These wines are represented by binary 0/1 patterns with $m = 90$ features, where each bit codes for the presence or absence of a specific feature. On average, every wine pattern has approx. $k = 14.7$ active units (one-entries denoting present features).

Each feature has a salience value that denotes its weight which is factored in the pairwise distance assessments of the hierarchical clustering procedure.

Given this dataset models domain knowledge, not surprisingly, the data are highly correlated (cf. Fig. 7). This is a highly desirable property, as it allows a meaningful organization to emerge from the clustering procedure. To choose an appropriate clustering method to produce a taxonomy for this dataset, we computed the Pearson correlation coefficient between pairwise taxonomical distances vis-à-vis the original pairwise distances in the data for the most common methods and distances, as presented in Table 5. We chose the combination that produced the highest correlational coefficient.

Due to the restriction of built-in semantics, our dataset is moderately sparse (cf. Fig. 7) by design (average activity level of any pattern is within approx. 16.4 % of the content space), that is, the data are not optimized for the strict sparseness requirement of a Willshaw network. Consequently, the network may overload and produce retrieval errors. Figure 8 depicts such a network after learning every pattern in the dataset with a Hebbian rule.

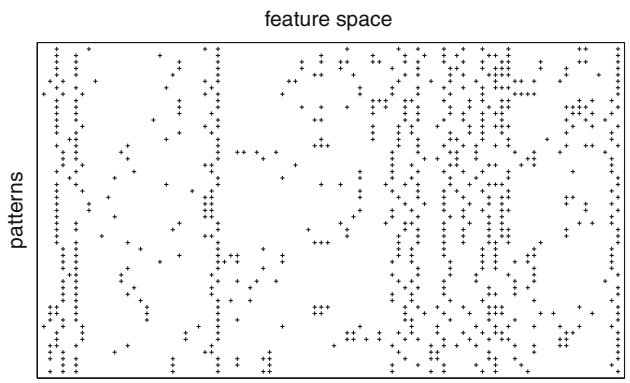


Fig. 7 Visual depiction of the binary wine dataset (the *dots* represent one-entries of the binary patterns). Feature distribution across the *x*-axis is not uniform, while along the *y*-axis, there is strong correlation between patterns

Table 5 Table of cophenetic correlation coefficients for varying clustering methods using pairwise Hamming (noted as *H*) and Jaccard (noted as *J*) distances applied to the wine dataset

	Cluster distance		
	Single link	Complete-link	Average-link
Normal			
H	0.509	0.646	0.672
J	0.671	0.750	0.779
Salient			
H	0.619	0.678	0.736
J	0.653	0.684	**0.795**

Coefficient values are presented for two settings: in the *Normal* scenario, no features are weighted, whereas in the *Salient* regime, a measure of feature salience is used when computing pairwise distances. Best value highlighted in bold

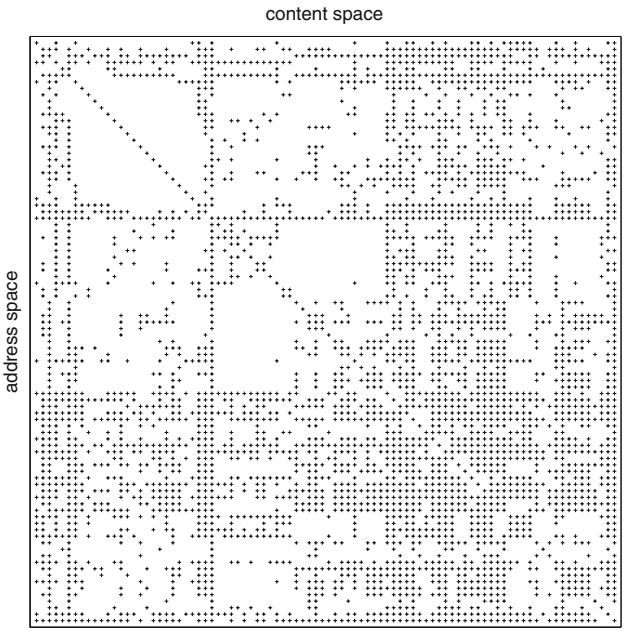


Fig. 8 Illustration of the cell assemblies formed in a Willshaw associative memory which has learned every pattern in the wine dataset. The *dots* represent active synapses. Note that synaptic connectivity is not uniform, reflecting the nature of the data where certain features are more distinctive

Concept Retrieval

A strong motivation of our model is to account for progressive retrieval where additional information is recalled at every step. Experiments with the wine dataset highlight this property. As shown in Table 6, after providing a cue to the memory, it produces a sequence of sets of shared

Table 6 Retrieval of a learned pattern describing a portuguese red wine from Setúbal, using an accurate query pattern $\tilde{\mathbf{x}} = \{$Red, Portugal, Setúbal, Blend, intense color, off dry, med. tannins, full body, dark fruits, red fruits, wood, long finish$\}$

Lv.		Concept feature-set $\mathscr{I}(C_p)$	#
2	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red$\}$	1
3	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red$\}$	1
4	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Blend$\}$	2
5	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Blend$\}$	2
6	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Blend, off dry,$\}$	3
7	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Blend, off dry, dark fruits, long finish$\}$	5
8	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Blend, intense color, off dry, full body, dark fruits, long finish$\}$	7
9	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Portugal, Blend, intense color, off dry, full body, dark fruits, long finish$\}$	8
10	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Portugal, Setúbal, Blend, intense color, off dry, med. tannins, full body, dark fruits, red fruits, wood, long finish$\}$	12

At taxonomical depth $d = 3$ and $d = 5$, the shared features of the matched clusters are precisely the same as at the previous levels ($d = 2$ and $d = 4$, respectively). Therefore, the superordinate concepts represented are the same

features, yielding an increasingly specific categorization, varying from abstract concepts described by small feature-sets toward increasingly specific concepts described by increasingly larger sets of shared features.

In principle, the most possible abstract concept representing *"everything"* contains all possible objects and is described by an empty set of shared features, whereas a concept described by a set containing every possible feature represents *"nothing"* (cf. theory of formal concepts as in [14, 76]). In practice, we found that at the higher levels of the taxonomies produced by our model, sometimes a concept encompasses such a diversity of elements that they share no common features. However, at these earlier levels (before any leaf nodes emerge), a taxonomy is essentially built by contrasting pairs of clusters: an element belongs to either a category (represented as a concept) or its contrasting sibling category. We can exploit the binary structure of the taxonomy to produce information about an element belonging to a category whose respective concept is too abstract to be described by a feature-set. In such case, it is possible to simply describe the feature-set of the sibling concept to which the element does not belong (cf. Table 7). This of course finds a limit if, at a given level, both contrasting concepts are too abstract for a prototypical description.

The grandmother-like positional coding used in the filter networks presented earlier may associate certain patterns with more than one cluster. The progressive retrieval and categorization of such a pattern is illustrated in Table 8.

Table 7 Retrieval of a learned pattern describing a portuguese white wine from the Dão region, with an accurate query pattern $\tilde{\mathbf{x}} = \{$White, Portugal, Dão, Blend, dry, med. body, citrus, med. finish$\}$

Lv.		Concept feature-set $\mathscr{I}(C_p)$	#
2	$\mathbf{x}^\mu \notin C_p$:	$\mathscr{I}(C_p) = \{$Red$\}$	1
3	$\mathbf{x}^\mu \notin C_p$:	$\mathscr{I}(C_p) = \{$Rosé, high acidity, red fruits$\}$	4
4	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$White$\}$	1
5	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$White, Blend$\}$	2
6	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$White, Portugal, Blend$\}$	3
...		...	
10	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$White, Portugal, Dão, Blend, dry, med. body, citrus, med. finish$\}$	8

At the last filter network ($d = 10$), the query produces a cluster representing a superordinate concept of $\tilde{\mathbf{x}}$ whose 8 prototypical features match 66.6% of $\tilde{\mathbf{x}}$. The clusters matched in earlier networks at depth $d \leq 3$ do not exhibit prototypical (shared) features; therefore, the description of their contrasting sibling clusters is shown

Table 8 Retrieval of a learned pattern describing a chilean wine, with an accurate query pattern $\tilde{\mathbf{x}} = \{$Red, Chile, Single Varietal, Cabernet Sauvignon, off dry, ...$\}$

Lv.		Concept feature-set $\mathscr{I}(C_p)$	#
2	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red$\}$	1
3	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Single Varietal$\}$	3
4	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Single Varietal dark fruits,...$\}$	4
5	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, South Africa,...$\}$	12
	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, full body, ...$\}$	6
...		...	
	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, South Africa, Single Varietal, Merlot, ...$\}$	12
7	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Argentina, Single Varietal, med. sweet, full body, ...$\}$	13
	$\mathbf{x}^\mu \in C_p$:	$\mathscr{I}(C_p) = \{$Red, Chile, Single Varietal, Cabernet Sauvignon, off dry, ...$\}$	14

At taxonomical depth $d = 7$, the retrieval process has identified three leaf nodes, one of which correctly matches $\tilde{\mathbf{x}}$

Performance Gains

The retrieval process in our model can be halted at any taxonomical level, using the results of the last intermediate network to filter the relevant units at the uncompressed associative memory. Alike the prime model, in developing a taxonomical model, we are concerned with improving the performance of the retrieval queries. We measure retrieval cost $\hat{\tau}$ as the number of operations needed to retrieve a pattern, that is, the number of neuron cells verified at every associative memory to satisfy a query. We express a performance ratio $\hat{\tau}/\tau$ where the cost of retrieving a pattern in our model is normalized by the cost of retrieving the same pattern using a single Willshaw network, that is, assessing

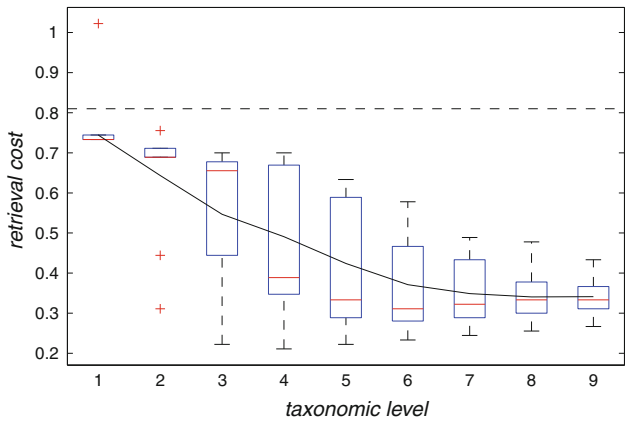


Fig. 9 Retrieval performance of a taxonomical memory which has learned every pattern of the wine dataset. Performance is measured in computation steps and normalized by the retrieval cost in a single Willshaw auto-associative memory. A *line plots* the evolution of the average retrieval cost of our model at varying taxonomical level, whereas a *dashed line marks* the retrieval cost achieved by the prime model with a configuration of a single filter network with aggregation factor $a_1 = 3$, determined by numerical optimization for this dataset

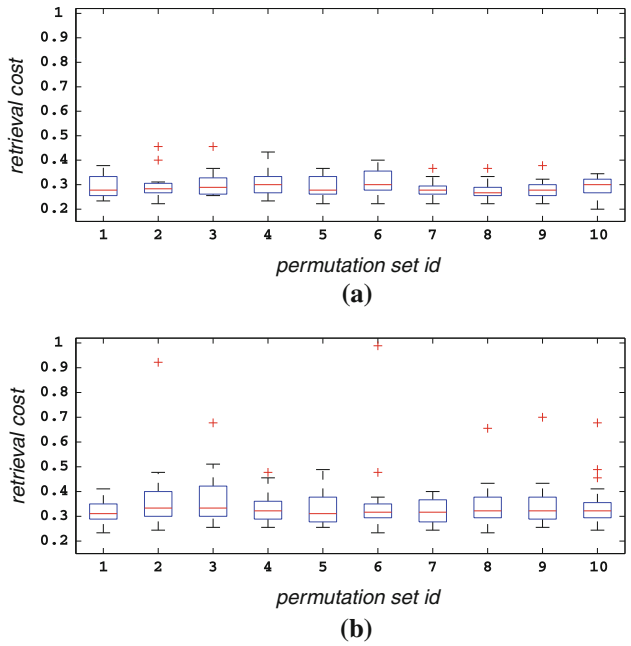


Fig. 10 Stability of retrieval cost performance at the optimal taxonomical level for random subsets of the wine dataset. **a** 10 random subsets of 20 out of 51 patterns. **b** 10 random subsets of 40 out of 51 patterns

the R-th memory without exploiting the taxonomical setup of intermediate memories. As shown in Fig. 9, the performance[2] of our model improves at every taxonomical level, that is, at each filter network. The prime model (dashed line) reduces average retrieval cost to approx. 81.6 % operations of a single Willshaw network, whereas our model is able to go further to approx. 35.2 % operations.

To assess the stability of the performance gains of our model, we tested taxonomical memories holding subsets of the wine dataset. The process was applied for 10 subsets with combinations of 20 out of 51 patterns, and another 10 subsets with combinations of 40 out of 51 patterns. For each subset, we considered the optimal taxonomical level that minimizes retrieval cost. The results[2] are shown in Fig. 10.

The lower quartile Q_1, the upper quartile Q_3, and especially the median of the performance ratio $\hat{\tau}_{\text{opt}}/\tau$ with $\hat{\tau}$ chosen at optimal taxonomical level are relatively stable for the varying subsets in both experiments, in spite of increasing outliers and slightly wider distributions for the combinations of 40 out of 51 patterns (cf. Fig. 10b).

Error Reduction

The wine dataset is only moderately sparse. Consequently, the R-th network in our model, being an auto-associative Willshaw network, will likely overload and operate with output distortion, even if the inputs are precise. However, each of our taxonomical filter networks aims to code one superordinate concept per pattern using a single bit per pattern. This grandmother-like coding warrants consistently low activity levels at every filter network and is able to reduce retrieval errors at the full uncompressed associative memory, caused by overloading.

We measure the error rate $\hat{\varepsilon}$ of retrieved patterns as the Hamming distance between cued and retrieved patterns normalized by cell assembly activity.

As shown in Fig. 11, our taxonomical memory has significative accuracy gains, attaining a zero error rate at the deepest level, whereas a hierarchical memory does not improve the error rate of the Willshaw memory.

Random Correlated Data

While the testing of our memory with a semantic dataset describes its categorization capabilities, we set out to generalize our experimental procedure with random data. Furthermore, we set out to produce random data with a fixed degree of sparseness, as well as a degree of hierarchical correlations, as to realistically mimic taxonomical data. Numerous studies have been conducted on the development of specialized learning rules for hierarchical correlated patterns and the storage capacity of these networks (see, e.g., [75, 77–81]).

[2] On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. Any datum higher/lower than 1.5 interquartile range ($Q_3 - Q_1$) of the lower/upper quartile is considered an outlier

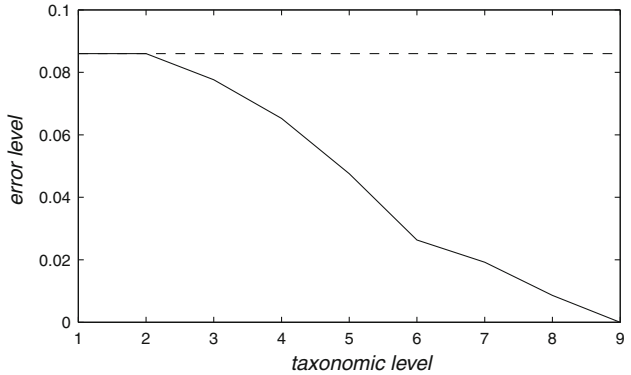


Fig. 11 Error reduction in a taxonomical memory holding all patterns of the wine dataset. Error rate is defined as the Hamming distance between cued and retrieved patterns normalized by cell assembly activity. A *solid line plots* the average retrieval error of our model, varying with taxonomical level, whereas the *dashes mark* the average error rate of a single uncompressed memory. The output distortion (add-error) at higher levels of the hierarchy is due to overloading, that is, the storage capacity of these memories is being stressed to its limits. Error rate decreases while pursuing the search at lower levels on account of two factors: (1) each network at a lower level uses more units to store the same number of associations stored previously, thus yielding higher capacity thresholds; (2) pattern retrieval is aided by the pruning operation that silences irrelevant units, thus reducing spurious firings

Artificial Correlated Datasets

To allow a comparison of the results of our experiments with random data and the results of the wine dataset, we controlled certain parameters for pattern generation to approximate the dimensions and properties of the wine dataset. Namely, we chose the following values: number of patterns $M = 50$, number of features $m = 100$, and rate of activity per pattern $f = 0.15$ (yielding $k = f \cdot m = 15$).

Random datasets with hierarchical correlations are generated in the manner of Kimoto and Okada [81]. First, we produced a smaller set of $s = \lceil 0.1M \rceil$ template patterns $\mathbf{x}^s$ using the parameters described above. For each of these templates, we consider a group $T = \lfloor M/s \rfloor$ of child patterns $\mathbf{x}^{st}$ where s is the number of the group of correlated patterns with the same parent pattern, and t identifies each pattern in a group. Each parent pattern $\mathbf{x}^s$ is a vector of m binary features, where each feature is denoted as x_i^s.

Memory patterns composing the correlated dataset are generated for every parent pattern, with each binary feature of a child pattern described by

$$P(x_i^{st} = 1) = \begin{cases} K : x_i^s = 1 \\ R : x_i^s = 0 \end{cases} \tag{22}$$

$i = 1, 2, \cdots, m$ and $t = 1, 2, \ldots, T$.

To assure that the activity level of each memory pattern $\mathbf{x}^{st}$ becomes f, we must set R to $f\,(1 - K)/(1 - f)$. In our experiments, we chose $K = 0.5$. For simplicity, we generate correlated data for our tests with only two hierarchical levels. However, the method could be recursively applied to produce deeper hierarchies of correlated patterns. By using fixed values for K (and consequently R), we are producing uniform correlations, that is, the degree of similarity between ancestors (first-level patterns) and descendants (second-level patterns) is fixed. More realistic correlations could be produced by varying K (and consequently R) along a range of parameters.

Trials with Correlated Data

For each trial, we produced two sets of randomly generated patterns: one contains purely random patterns, and another contains hierarchically correlated random patterns. To provide a baseline for our model, at each trial, each set was stored and retrieved from an auto-associative Willshaw network. We computed average retrieval cost τ and average retrieval error ε over each set. Not surprisingly, a simple Willshaw network produces a higher error rate ε for correlated patterns as shown in Fig. 12.

Afterward, the sets were stored and retrieved from a taxonomical associative memory. Over each set, we measured the average retrieval cost $\hat{\tau}$ and average retrieval error $\hat{\varepsilon}$ for all taxonomical levels of these memories and determined the optimal taxonomical levels d_τ and d_ε that minimize $\hat{\tau}$ and $\hat{\varepsilon}$, respectively. Let us then denote average retrieval cost at taxonomical level d_τ as $\hat{\tau}_{\text{opt}}$ and average retrieval error at taxonomical level d_ε by $\hat{\varepsilon}_{\text{opt}}$.

Figure 13a presents the distribution of performance gains (expressed by the ratio $\hat{\tau}_{\text{opt}}/\tau$) for random and for correlated datasets. Figure 13b presents the analogous comparison for the distribution of accuracy gains (expressed by the ratio $\hat{\varepsilon}_{\text{opt}}/\varepsilon$).

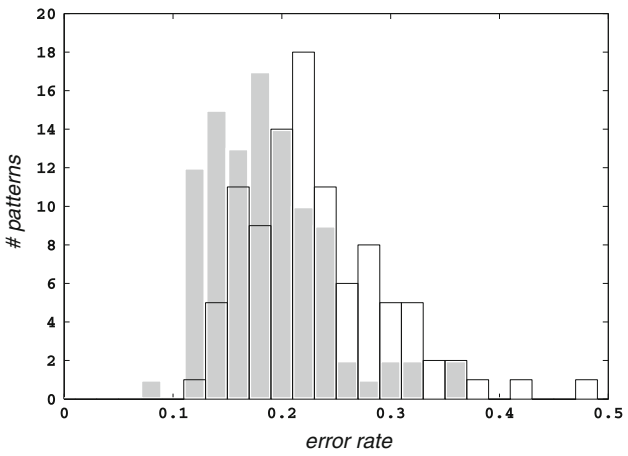
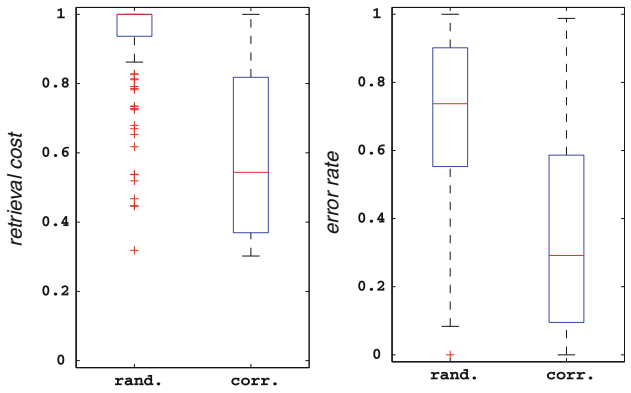


Fig. 12 Distribution of average retrieval error ε of a single network for random (*filled-bars*) versus correlated (*outlined bars*) patterns over 100 trials. Average error rate for hierarchically correlated sets is higher than for purely random datasets



(a) Performance gain $\hat{\tau}_{opt}/\tau$ **(b)** Accuracy gain $\hat{\varepsilon}_{opt}/\varepsilon$

Fig. 13 Distribution of average performance and accuracy gain of a taxonomical memory for random versus correlated patterns over 100 trials

As expected, a taxonomical associative memory attains best results with hierarchically correlated patterns. It is so for optimal cost reduction as well as optimal error reduction.

Conditioned Optimization of Accuracy and Performance

To better understand the interplay of accuracy versus performance gains, we considered an additional analysis where we observe the frequency of accuracy and performance ratios over a series of trials with correlated data.

Let us denote conditioned retrieval cost $\hat{\tau}_{cond}$ as the minimal retrieval cost $\hat{\tau}$ obtained at the optimal taxonomical level d_ε that minimizes retrieval error $\hat{\varepsilon}$. Conditioned performance ratio $\hat{\tau}_{cond}/\tau$ expresses the conditioned retrieval cost normalized by the retrieval cost τ of a Willshaw network for the same data.

Conversely, conditioned error rate $\hat{\varepsilon}_{cond}$ is defined as the minimal error rate $\hat{\varepsilon}$ obtained at the optimal taxonomical level d_τ that minimizes retrieval cost $\hat{\tau}$. Conditioned accuracy ratio $\hat{\varepsilon}_{cond}/\varepsilon$ expresses the conditioned error rate normalized by the error rate ε of a Willshaw network for the same data.

Figure 14 presents the distribution of conditioned accuracy ratio versus optimal performance ratio. Two *foci* are observable in this distribution: one where $\hat{\tau}_{opt}/\tau \approx 0.3$ and $\hat{\varepsilon}_{opt}/\varepsilon \approx 0$ representing the best scenario where both goals are optimized, and another where $\hat{\tau}_{opt}/\tau \approx 1$ and $\hat{\varepsilon}_{opt}/\varepsilon \approx 1$ where no performance gain is possible, hence, no error reduction is attempted.

On the other hand, Fig. 15 presents the distribution of conditioned performance ratio versus optimal accuracy ratio. Two *foci* are also easily distinguishable in this distribution: one where $\hat{\tau}_{opt}/\tau \approx 0.3$ and $\hat{\varepsilon}_{opt}/\varepsilon \approx 0$ representing the best scenario where both goals are optimized, and another where $\hat{\tau}_{opt}/\tau \approx 1.2$ and $\hat{\varepsilon}_{opt}/\varepsilon \approx 0.8$ where a

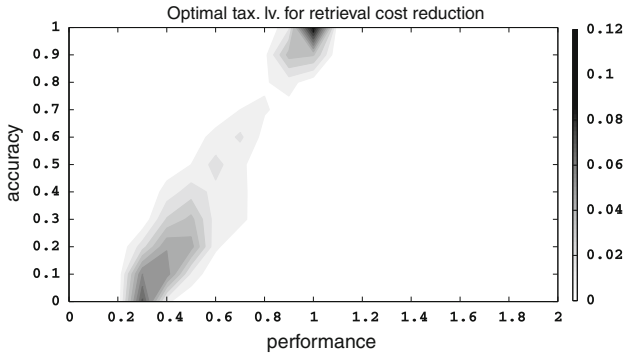


Fig. 14 Distribution of average accuracy versus performance ratios at optimal level d_τ over 1,000 trials with correlated data

modest error reduction is achievable at the expense of additional retrieval costs.

Thus, our model allows two possible retrieval regimes. By optimizing the taxonomical level of retrieval for performance, one allows accuracy gains to emerge, whereas if retrieval is tuned for accuracy, one accepts to incur on additional retrieval costs if it yields any accuracy gain, no matter how small.

Discussion

Motivated by biophysical and psychological evidence from cognitive studies of categorization, we have developed a taxonomical associative memory model that employs Willshaw nets at its core. These nets, while computationally simple, provide a mechanism of synaptic plasticity that is biologically sound.

We have shown that in a hierarchical configuration of associative networks with taxonomical semantics, learned patterns can be retrieved progressively while producing descriptions of superordinate concepts. Moreover, alike the hierarchical retrieval prescription described in Sacramento and Wichert [1], our taxonomical model yields a performance improvement over the original Willshaw network.

We have also found that in a high memory load regime, while a simple Willshaw network tuned for pattern completion (global threshold $\Theta = |\tilde{\mathbf{x}}|$) produces add-errors, our model (configured with the same Θ) is able to reduce these errors. Our model employs compressed associative networks which improve the performance (neural firings) and accuracy (denoising) of retrieval at an uncompressed associative network under heavy load.

We have observed that the optimal taxonomical level R^{opt}, that is, the number of cell assemblies to activate during retrieval, is dependent on the optimization goal. Whether we optimize for performance or accuracy, gains are observed for the majority of recalled traces. Furthermore, we observed that certain memory traces can be

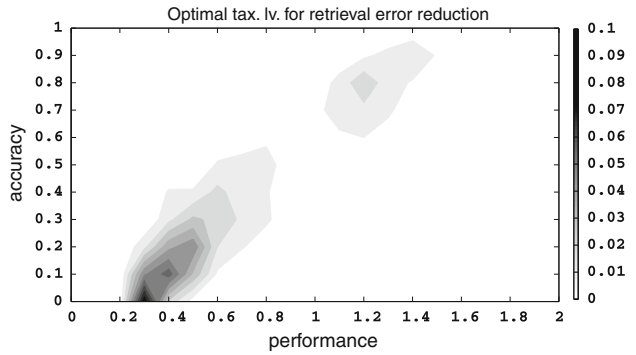


Fig. 15 Distribution of average accuracy versus performance ratios at optimal level d_ε over 1,000 trials with correlated data

retrieved with higher accuracy if we allow a penalty on performance by recruiting auxiliary assemblies for the progressive retrieval task.

These simulations are consistent with the idea that cognitive economies reward semantic organization of the data, and support the notion that increased neural activation yields increased precision of recall.

As in most of the classical associative memory models (e.g., [11, 74]) that rely on Hebbian processes, in our model, the interplay of accuracy and performance depends on the correlational structure of the data. We have included a non-local preprocessing step where a clustering process decorrelates the data and produces a taxonomic cluster-tree of variable shape (namely branching factor and heights) whose semantic groups are then encoded in a set of associative nets.

We hypothesize that the nature of the correlations in the input data (controlled by a set of parameters in the case of the artificial dataset of "Random Correlated Data") and the structure of the taxonomy resulting from the clustering process influence the denoising phenomena.

Biological Fidelity and Implications

We extended a theoretical model of hierarchical associative memory in an attempt to model psychological phenomena regarding the cognitive economies provided by taxonomic organization of semantic knowledge. We predict that different cell assemblies may correspond to different conceptual hierarchies and may be located at different brain regions. The hierarchical process of categorization would correspond to the sequential activation of different assemblies in different regions, progressively obtaining a refined resolution memory trace. This process of hierarchic retrieval by progressive categorization (generic-specialization IS-A relationships) could also be related to synfire chain theory [82, 83] which has been suggested to explain hierarchic organization of semantic knowledge for composition processes (whole-part HAS-A relationships) [84].

Imaging studies provide hints on cortical activation during categorization (e.g., within-category similarity is associated with the fusiform gyrus); however, it is still not completely clear how factors such as stimulus modality, inference, and contextual mechanisms are integrated. Given the fact that these factors are out of the scope of our model, it would be difficult to provide a solid link to experimental studies on these matters; thus, we refrain from speculation on neuronal localization.

Generalization and Priming

Although category membership is determined by the clustering process, we chose to code overlapping feature-set partitions to reduce learning time (see Appendix A). An interesting byproduct of this minor optimization is that patterns that were not clustered and that only share features of overlapping partitions may be associated with more than one category at every level, and yet, this coding aids in retrieval processing. The exploitation of overlapping feature-set partitions could be an avenue for future work allowing for fuzzy categorization, and these dynamics could account for uncertain categorization and priming phenomena.

Priming during retrieval could be modeled with a heuristic function estimating frequency of a category as proposed in earlier work [26]. This function would work as a reinforcement learning mechanism by strengthening links between categories on the basis of a smaller measure of error of the retrieval process when deciding between categories. After successive recall attempts, the categories which are primed at every taxonomical level take precedence whenever there is uncertainty regarding the categorization of an item.

Storage Efficiency

Our progressive retrieval prescription allows error reduction in overloaded memories while producing descriptions of progressively specific superordinate concepts. To achieve this, we have introduced complexity into the model which is in stark contrast with the biological perspective.

The added functionality implies an overhead in storage costs that, while negligible in a computer, represent a number of additional synaptic contacts organized in an unorthodox and simplistic topology considering what is known of the neuronal organization of the mammalian brain.

To allow the possibility of halting the retrieval process at any taxonomical rank, we must preserve, for every cluster, a feature-set describing the union of features of all its elements $\mathscr{U}(C_p)$, and to allow the description of the respective superordinate concept, we must also keep a feature-set describing the intersection of said features $\mathscr{I}(C_p)$.

Can this information be otherwise efficiently coded or economically stored? A possibility would be to use a recurrent auto-associative network for each cluster C_p, configured as an oscillator with two fixed attractor states: $\mathscr{U}(C_p)$ and $\mathscr{I}(C_p)$ [14]. If we are able to reduce the number of clusters formed during learning and increase the number of elements per cluster, we could model hierarchical concept formation with a biologically plausible hierarchy of oscillator networks. To reduce the number of oscillators needed, we shall exploit the heights of the hierarchical clustering to flatten the taxonomy and thus produce less clusters containing more elements.

Online Learning

An important aspect of our model that is at odds with the biological angle is the clustering procedure applied at the learning stage to elicit correlations in the data. This process is done in batch, and the resulting taxonomy is only valid for the set of patterns fed to the clustering. To make this step biologically sound, a similar procedure must be used which is both distributed—that is, works with local rules as expected of a cell assembly—and on-line—that is, deals with continuous input, and iteratively adjusts the taxonomical tree. Study of diverse learning models could lead to a neural mechanism able to efficiently code input data with hierarchical correlations, yielding a distributed, local and sparse taxonomical code. We propose avenues of research on this topic and a commentary on related models.

To enable our model for continuous learning, it must be extended with some capability for generalization and self-organization of categories. More specifically, we must abandon our bottom-up agglomerative approach in favor of a continuous statistical process of coincidence detection designed to extract prototypes from an open flow of examples and to arrange these prototypes in a hierarchical fashion.

Such a generalization system could be based on a competitive learning Hebbian rule, for example, winner-take-all, in the manner of Kohonen's self-organizing maps [63].

Alternatively, another competitive strategy would employ a sparse coding scheme alike Foldiak's cell assembly models which use anti-Hebbian learning [85] to code sparse representations where statistical dependencies present in the input layer are reduced, that is, such system proposes to code objects as groups of highly correlated sets of components that are relatively independent from other such groups; this maps nicely to our notions of within-category feature sharedness versus inter-category feature distinctiveness. Investigation into the stacking of such layers would be required to assess the possibility of building a hierarchy of decorrelation mechanisms to act as our taxonomic tree.

Finally, an interesting proposal would be to base this system on palimpsest models of slow learning [86]. These models feature discrete synapses and have been shown to produce prototypes given a continuous stream of exemplars. Moreover, in this type of model, the prototypes are continuously adjusted as traces of older exemplars are forgotten.

Independently of the direction taken to achieve a biologically sound learning mechanism based on prototype extraction, a major challenge relies in the integration of said mechanism in our model. The produced prototypes (representing categories) must be generated in a way such that their semantic aggregates (i.e., the Boolean sum patterns) overlap at a minimum in order to maintain the filtering capabilities of our model that allow for error reduction and performance improvement.

Technical Improvements

Aside from the semantic aspects of the taxonomical model, we have shown how it improves on technical aspects of a simple associative memory; namely by its error-reducing capabilities for a regime of high memory load and moderate sparseness (cf. load tolerance of a Willshaw network). However, associative memory models are specially interesting for their resilience to input distortion, namely miss-noise. In an auto-associative setup, if a sparse requirement is met, these memories perform efficient pattern completion. On the other hand, add-noise is more problematic. The grandmother coding inherent to our model is particularly susceptible to add-noise as it is a very simple code.

The resilience of the taxonomical associative memory to input distortion can be improved in regard to both miss-noise and add-noise by adding redundancy to the grandmother coding.

For a comprehensive study of storage capacity, our model should be further tested with more complex learning rules such as the covariance rule [87–89] or the more recent optimal Bayesian rule [90], which demand a weaker sparse requirement from the data and allow higher memory load.

On another note, as reviewed in "Mental Representations", several studies have been conducted [75, 77–79] on the capabilities of saturated Hopfield networks to generalize concepts by pattern completion if enough exemplars of a concept were previously learned. These models employ learning rules which demand prior knowledge of the correlational structure in the data, whereas our model works independently of this structure. Nevertheless, our taxonomical model should be further analyzed in comparison with these *categorization by generalization* models.

Acknowledgments This work was supported by national funds through FCT—Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2011. J.S. is supported by an FCT doctoral grant (contract SFRH/BD/66398/2009).

Appendix: Cluster Membership Heuristic

After the preprocessing step where elements are clustered, learning of item–category membership associations is performed at every taxonomical rank by an associative net. To determine the cluster C_p that contains a certain item x at a certain taxonomical rank, we could simply refer to our cluster-tree.

Alternatively, drawing inspiration from the fixed-points of the oscillator model presented in [14], we propose a heuristic to test whether an element belongs to a cluster which warrants no false negatives (which would impair accuracy). This approach reduces lookup costs in the computational setting and provides a hint of how the model could generalize learning while leveraging a symbolic preprocessing step.

We use a convenient short-hand representation of a cluster, the Boolean sum pattern $\mathscr{U}(C_p)$, previously presented in eq. 20, that describes the feature-set containing every feature which is present in at least one element of C_p. Once again we turn to our simple fruit taxonomy to illustrate. Table 9 shows these feature-sets $\mathscr{U}(C_p)$ for each non-single-element cluster. For a single-element cluster, this feature-set is given by that very element.

These feature-sets are analogous to the extension of the Boolean OR aggregates in the prime model. Note, however, that unlike the prime model, the content space is not partitioned, that is, the divisions are possibly overlapping. Thus, in our model, two contrasting feature-sets (at the same taxonomical level) may have a nonzero intersection. For instance, in our illustrative taxonomy of fruits (refer to Table 9), we have $C_2 \cap C_3 = \{\text{sweet, round}\}$.

Given the requirement that we produce no false negatives when pruning from one network to the next, we defined our pattern versus cluster matching heuristic to require a single shared feature between a pattern x and a cluster description $\mathscr{U}(C_p)$ to associate a pattern with said cluster C_p.

Table 9 Fruit clusters: all features

Cluster	$\mathscr{U}(C_p)$
C_1	{sour, sweet, hard, round, citrus juicy}
C_2	{sour, sweet, round, citrus, juicy}
C_3	{sweet, hard, round}
C_4	{sour, round, citrus, juicy}

Single-element clusters are not represented

Table 10 Table presenting item–category associations stored at every auxiliary network for the fruit dataset

Fruit $\mathbf{x}^\mu$	Lemon 010011	Lime 011011	Orange 101011	Apple 101100	Plum 101000
1^{st} *net*	C_2	$C_2 \vee C_3$	$C_2 \vee C_3$	$C_2 \vee C_3$	$C_2 \vee C_3$
$\xi_2(\mathbf{x}^\mu)$	10	11	11	11	11
2^{nd} *net*	$C_4 \vee C_5$	$C_4 \vee C_5 \vee C_6 \vee C_7$	$C_4 \vee C_5 \vee C_6 \vee C_7$	$C_5 \vee C_6 \vee C_7$	$C_5 \vee C_6 \vee C_7$
$\xi_3(\mathbf{x}^\mu)$	1100	1111	1111	0111	0111
3^{rd} *net*	$C_5 \vee C_8 \vee C_9$	$C_5 \vee C_6 \vee C_7 \vee C_8 \vee C_9$	$C_5 \vee C_6 \vee C_7 \vee C_8 \vee C_9$	$C_5 \vee C_6 \vee C_7$	$C_5 \vee C_6 \vee C_7$
$\xi_4(\mathbf{x}^\mu)$	11100	11111	11111	00111	00111

Fruits are coded for feature presence following the order of the feature-set {sweet, sour, round, hard, citrus, juicy}. Note that we consider $\delta = 2$, meaning the first auxiliary network codes for $d = 2$, distinguishing C_2 from C_3

We exploit the binary structure of the taxonomical tree, testing always cluster pairs. Alike a binary search procedure, when we are checking at which clusters an element $\mathbf{x}^\mu$ belongs, if we have determined that at a given level $\mathbf{x}^\mu \in C_a \wedge \mathbf{x}^\mu \notin C_b$, we may at deeper levels disregard the descendants of C_b.

This approach carries a trade-off: it produces false positives. Consider, for instance, the definition of *apple* in our fruit taxonomy (refer to Table 1 and the feature-sets of C_2 and C_3 in Table 9). According to our heuristic, we cannot determine which of these clusters contain the pattern (both produce feature matches); however, at deeper levels, this uncertainty of "taxonomical location" decreases.

For an illustration of the side-effects of this method, consider Table 10 depicting item–category associations for the fruit taxonomy where cluster codes are approximated with the heuristic here described. Given the compact and dense feature space of this dataset, and the fact it produces a taxonomy that is not very deep, overlaps in feature-unions are plenty and reduction in uncertainty is minimal.

References

1. Sacramento J, Wichert A. Tree-like hierarchical associative memory structures. Neural Netw. 2011;24(2):143–7.
2. Harnad S. To cognize is to categorize: cognition is categorization. Handbook of categorization in cognitive science. 2005. pp. 19–43.
3. Rosch E. Principles of categorization. In: Rosch E, Lloyd BB, editors. Cognition and categorization. Hillsdale, NJ: Lawrence Erlbaum Associates; 1978. p. 27–48. (Reprinted in *Readings in Cognitive Science. A Perspective from Psychology and Artificial Intelligence*, A. Collins and E.E. Smith, editors, Morgan Kaufmann Publishers, Los Altos (CA), USA, 1991).

4. Berlin B. Ethnobiological classification: principles of categorization of plants and animals in traditional societies. Princeton, NJ: Princeton University Press; 1992.
5. Caramazza A, Shelton JR. Domain-specific knowledge systems in the brain: the animate-inanimate distinction. J Cogn Neurosci. 1998;10(1):1–34.
6. Warrington EK, McCarthy R. Category specific access dysphasia. Brain 1983;106(4):859–78.
7. Perani D, Schnur T, Tettamanti M, Cappa SF, Fazio F, et al. Word and picture matching: a PET study of semantic category effects. Neuropsychologia 1999;37(3):293–06.
8. Thompson-Schill S, Aguirre G, Desposito M, Farah M. A neural basis for category and modality specificity of semantic knowledge. Neuropsychologia 1999;37(6):671–6.
9. Ishai A, Ungerleider LG, Martin A, Schouten JL, Haxby JV. Distributed representation of objects in the human ventral visual pathway. Proc Natl Acad Sci. 1999;96(16):9379.
10. Sacramento J, Burnay F, Wichert A. Regarding the temporal requirements of a hierarchical Willshaw network. Neural Networks. 2012;25:84–93. doi:10.1016/j.neunet.2011.07.005.
11. Willshaw DJ, Buneman OP, Longuet-Higgins HC. Non-Holographic Associative Memory. Nature. 1969 06;222(5197):960–962.
12. Palm G. On associative memory. Biol Cybern. 1980;36:19–31. doi:10.1007/BF00337019.
13. Palm G. Towards a theory of cell assemblies. Biol Cybern. 1981;39:181–94. doi:10.1007/BF00342771.
14. Wennekers T. On the natural hierarchical composition of cliques in cell assemblies. Cogn Comput. 2009;1:128–38.
15. Apostle HG. Aristotle's Categories and propositions (De Interpretatione). Grinnell, IA: Peripatetic Press; 1980.
16. Murphy GL. The big book of concepts. Cambridge: MIT Press; 2002.
17. Smith EE, Medin DL. Categories and concepts. In: Smith EE, Medin DL, editors. Harvard University Press, Cambridge, MA; 1981.
18. Barsalou LW. Ideals, central tendency, and frequency of instantiation as determinants of graded structure in categories. J Exp Psychol Learn Memory Cogn. 1985;11(4):629–54.
19. Rosch E, Mervis CB, Gray WD, Johnson DM, Boyes-Braem P. Basic objects in natural categories. Cogn Psychol. 1976;8(3):382–439.
20. Smith EE. Concepts and categorization. In: Osherson EESD, editor. Thinking. vol. 3. 2nd ed. Cambridge, MA: MIT Press; 1995. pp. 3–33.
21. McClelland JL, Rumelhart DE. Distributed memory and the representation of general and specific information. J Exp Psychol Gen. 1985;114(2):159–88.
22. Tversky A. Features of similarity. Psychol Rev. 1977;84(4):327–52.
23. Osherson DN. Probability judgement. In: Osherson EESD, editor. Thinking. vol. 3. 2nd ed. Cambridge, MA: MIT Press; 1995. pp. 35–75.
24. Rosch E, Mervis CB. Family resemblances: studies in the internal structure of categories. Cogn Psychol. 1975;7(4):573–605.
25. Rips LJ, Shoben EJ, Smith EE. Semantic distance and the verification of semantic relations. J Verbal Learn Verbal Behav. 1973;12(1):1–20.
26. Wichert A. A categorical expert system "Jurassic". Expert Syst Appl. 2000;(19):149–58.
27. Nosofsky RM. Attention, similarity, and the identification-categorization relationship. J Exp Psychol Gen. 1986;115(1):39–61.
23. Kurtz DGK. Relational Categories. In: Ahn WK, Goldstone RL, Love BC, Markman AB, Wolff PW, editors. Categorization inside and outside the lab. Washington, DC: American Psychological Association; 2005. pp. 151–175.
29. Waltz J, Lau A, Grewal S, Holyoak K. The role of working memory in analogical mapping. Memory Cogn. 2000;28:1205–12. doi:10.3758/BF03211821.
30. Smith EE, Grossman M. Multiple systems of category learning. Neurosci Biobehav Rev. 2008;32(2):249–64. (The Cognitive Neuroscience of Category Learning).
31. Tomlinson M, Love B. When learning to classify by relations is easier than by features. Think Reason. 2010;16(4):372–401.
32. Doumas LAA, Hummel JE, Sandhofer CM. A Theory of the discovery and predication of relational concepts. Psychol Rev. 2008;115(1):1–43.
33. Kay P. Taxonomy and semantic contrast. Language. 1971;47(4):866–887.
34. Murphy GL, Brownell HH. Category differentiation in object recognition: typicality constraints on the basic category advantage. J Exp Psychol Learn Memory Cogn. 1985;11(1):70–84.
35. Sneath PHA, Sokal RR. Numerical taxonomy. Nature. 1962 03;193(4818):855–0.
36. Sneath PH. The application of computers to taxonomy. J Gen Microbiol. 1957;17:201–26.
37. Jaccard P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin del la Société Vaudoise des Sciences Naturelles. 1901;37:547–79.
38. Hastie T, Tibshirani R, Friedman JH. The elements of statistical learning, Corrected ed. Springer, Berlin; 2003.
39. Tan PN, Steinbach M, Kumar V. Introduction to data mining, used ed. Addison Wesley, Reading, MA; 2005.
40. Manning CD, Raghavan P, Schütze H. Introduction to information retrieval, 1st ed. Cambridge University Press, Cambridge; 2008.
41. Sokal RR. Numerical taxonomy. Sci Am. 1966;215(6):106–116.
42. Anderson JR, Bower GH. Human associative memory. Winston, Washington; 1973.
43. Collins A, Quillian M. Retrieval time from semantic memory. J Verbal Learn Verbal Behav. 1969;8(2):240–7.
44. Rumelhart DE, McClelland JL. Parallel distributed processing: explorations in the microstructure of cognition, vol 1: foundations. MIT Press, Cambridge, MA; 1986.
45. Collins AM, Loftus EF. A spreading-activation theory of semantic processing. Psychol Rev. 1975;82(6):407–28.
46. Rojas R. Neural networks: a systematic introduction. Springer, Berlin; 1996.
47. Steinbuch K. Die Lernmatrix. Kybernetic 1961;1:36–45.
48. Amari SI. Characteristics of sparsely encoded associative memory. Neural Netw. 1989;2(6):451–7.
49. Nadal JP, Toulouse G. Information storage in sparsely coded memory nets. Netw Comput Neural Syst. 1990;1(1):61–74.
50. Buckingham J, Willshaw D. Performance characteristics of the associative net. Netw Comput Neural Syst. 1992;3(4):407–14.
51. Graham B, Willshaw D. Improving recall from an associative memory. Biol Cybern. 1995;72(4):337–46.
52. Knoblauch A, Palm G, Sommer FT. Memory capacities for synaptic and structural plasticity. Neural Comput. 2010;22(2):289–41.
53. Hebb DO. The organization of behaviour. Wiley, New York; 1949.
54. Buckingham J, Willshaw D. On setting unit thresholds in an incompletely connected associative net. Netw Comput Neural Syst. 1993;4(4):441–59.
55. Schwenker F, Sommer FT, Palm G. Iterative retrieval of sparsely coded associative memory patterns. Neural Netw. 1996;9(3):445–55.
56. Wichert A. Subspace tree. In: IEEE on seventh international workshop on content-based multimedia indexing conference proceedings, 2009; p. 38–43.

57. Reed SK. Pattern recognition and categorization. Cogn Psychol. 1972;3(3):382–07.
58. Jones GV. Identifying basic categories. Psychol Bull. 1983;94(3): 423.
59. Edgell SE. Using configural and dimensional information. Individual and group decision making: current issues; 1993. p. 43.
60. Gluck M, Corter J. Information, uncertainty, and the utility of categories. In: Proceedings of the seventh annual conference of the cognitive science society. Hillsdale, NJ: Erlbaum; 1985. pp. 283–287.
61. Rosenblatt F. Principles of neurodynamics: perceptrons and the theory of brain mechanisms. Washington DC: Spartan; 1962.
62. Rumelhart DE, Hintont GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323(6088):533–6.
63. Kohonen T. Self-organized formation of topologically correct feature maps. Biol Cybern. 1982;43(1):59–9.
64. Waibel A, Hanazawa T, Hinton G, Shikano K, Lang KJ. Phoneme recognition using time-delay neural networks. IEEE Trans Acoustics Speech Signal Proc. 1989;37(3):328–39.
65. Cohen LB, Chaput HH, Cashon CH. A constructivist model of infant cognition. Cogn Dev. 2002;17(3):1323–43.
66. Brunel N. Storage capacity of neural networks: effect of the fluctuations of the number of active neurons per memory. J Phys A Math Gen. 1994;27(14):4783–9.
67. Petersen CCH, Malenka RC, Nicoll RA, Hopfield JJ. All-or-none potentiation at CA3-CA1 synapses. Proc Natl Acad Sci. 1998;95(8):4732–7.
68. O'Connor DH, Wittenberg GM, Wang SSH. Graded bidirectional synaptic plasticity is composed of switch-like unitary events. Proc Natl Acad Sci USA. 2005;102(27):9679–4.
69. Amit DJ, Fusi S. Learning in neural networks with material synapses. Neural Comput. 1994;6(5):957–82.
70. Fusi S, Abbott LF. Limits on the memory storage capacity of bounded synapses. Nature Neurosci. 2007;10(4):485–493.
71. Barrett AB, van Rossum MCW. Optimal learning rules for discrete synapses. PLoS Comput Biol. 2008 11;4(11):e1000230.
72. Leibold C, Kempter R. Sparseness constrains the prolongation of memory lifetime via synaptic metaplasticity. Cerebral Cortex 2008;18(1):67–7.
73. Huang Y, Amit Y. Capacity analysis in multi-state synaptic models: a retrieval probability perspective. J Comput Neurosci. 2011;30(3):699–20.
74. Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci USA. 1982;79(8):2554–58.
75. Gutfreund H. Neural networks with hierarchically correlated patterns. Phys Rev A 1988;37(2):570–7.
76. Belohlávek R. Representation of concept lattices by bidirectional associative memories. Neural Comput. 2000;12:2279–90.
77. Parga N, Virasoro MA. The ultrametric organization of memories in a neural network. J Phys. 1986;47(11):1857–64.
78. Toulouse G, Dehaene S, Changeux JP. Spin glass model of learning by selection. Proc Natl Acad Sci. 1986;83(6):1695–8.
79. Fontanari JF. Generalization in a Hopfield network. J Phys France 1990;51(21):2421–0.
80. Engel A. Storage of hierarchically correlated patterns. J PhysA Math Gen. 1990;23:2587.
81. Kimoto T, Okada M. Coexistence of memory patterns and mixed states in a sparsely encoded associative memory model storing ultrametric patterns. Biol Cybern. 2004;90(4):229–38.
82. Abeles M. Local cortical circuits: an electrophysiological study. Springer, New York; 1982.
83. Abeles M. Corticonics: neural circuits of the cerebral cortex. Cambridge University Press, Cambridge; 1991.
84. Abeles M, Hayon G, Lehmann D. Modeling compositionality by dynamic binding of synfire chains. J Comput Neurosci. 2004; 17(2):179–01.
85. Földiák P. Forming sparse representations by local anti-Hebbian learning. Biol Cybern. 1990;64(2):165–0.
86. Brunel N, Carusi F, Fusi S. Slow stochastic Hebbian learning of classes of stimuli in a recurrent neural network. Netw Comput Neural Syst. 1998;9(1):123–52.
87. Sejnowski TJ. Storing covariance with nonlinearly interacting neurons. J Math Biol. 1977;4(4):303–21.
88. Amit DJ, Gutfreund H, Sompolinsky H. Information storage in neural networks with low levels of activity. Phys Rev A. 1987;35(5):2293–303.
89. Dayan P, Willshaw DJ. Optimising synaptic learning rules in linear associative memories. Biol Cybern. 1991;65(4):253–65.
90. Knoblauch A. Neural associative memory with optimal Bayesian learning. Neural Comput. 2011;23(6):1393–451.