Neural Sub-Symbolic Reasoning

Andreas Wichert
Department of Informatics
INESC-ID / IST - Technical University of Lisboa
Portugal
andreas.wichert@ist.utl.pt

Abstract

The sub-symbolical representation often corresponds to a pattern that mirrors the way the biological sense organs describe the world. Sparse binary vectors can describe sub-symbolic representation, which can be efficiently stored in associative memories. According to the production system theory, we can define a geometrically based problem-solving model as a production system operating on sub-symbols. Our goal is to form a sequence of associations, which lead to a desired state represented by sub-symbols, from an initial state represented by sub-symbols. We define a simple and universal heuristics function, which takes into account the relationship between the vector and the corresponding similarity of the represented object or state in the real world.

1 Introduction

One form of distributed representation corresponds to a pattern that mirrors the way the biological sense organs describe the world. Sense organs sense the world by receptors. By the given order of the receptors the living organisms experience the reality as a simple Euclidian geometrical world. Changes in the world correspond to the changes in the distributed representation. Prediction of these changes by the nervous system corresponds to a simple geometrical reasoning process. Mental imagery problem solving is an example for a complex geometrical problem- solving. It is described by a sequence of associations, which progressively change the mental imagery until a desired solution of a problem is formed. For example, do the skis fit in the boot of my car? Mental representations of images retain the depictive properties of the image itself as perceived by the eye [Kosslyn, 1994]. The imagery is formed without perception through the construction of the represented object from memory. Symbols on the other hand are not present in the world; they are the constructs of human mind to simplify the process of problem solving. Symbols are used to denote or refer to something other than them, namely other things in the world (according to the pioneering work of Tarski [Tarski, 1956]). They are defined by their occurrence in a structure and by a formal language, which manipulates these structures [Simon, 1991; Newell, 1990]. In this context, symbols do not by themselves, represent any utilizable knowledge. They cannot be used for a definition of similarity criteria between themselves. The use of symbols in algorithms which imitate human intelligent behavior led to the famous physical symbol system hypothesis by Newell and Simon (1976) [Newell and Simon, 1976]: The necessary and sufficient condition for a physical system to exhibit intelligence is that it be a physical symbol system. We do not agree with the physical symbol system hypothesis. Instead we state that the actual perception of the world and manipulation in the world by living organisms lead to the invention or recreation of an experience, which at least in some respects, resembles the experience of actually perceiving and manipulating objects in the absence of direct sensory stimulation. This kind of representation is called sub-symbolic. Sub-symbolic representation implies heuristic functions. Symbols liberate us from the reality of the world although they are embodied in geometrical problem solving through the usage of additional heuristics functions. Without the use of heuristic functions real world problems become intractable.

The paper is organized a follows: We review the representation principles of objects by features as used in cognitive science. In the next step we indicate how the perception-oriented representation is build on this approach. We define the sparse sub-symbolical representation. Finally, we will introduce the sub-symbolical problem solving which relies on a sensorial representation of the reality.

2 Sub-symbols

Perception-oriented representation is an example of sub-symbolical representation, such as the representation of numbers by the Oksapmin tribe of Papua New Guinea. The Oksapmin tribe of Papua New Guinea counts by associating a number with the position of the body [Lancy, 1983]. The sub-symbolical representation often corresponds to a pattern that mirrors the way the biological sense organs describe the world. Vectors represent patterns. A vector is only a sub-symbol if there is a relationship between the vector and the corresponding similarity of the represented object or state in the real world through sensors or biological senses. Feature based representation is an example of sub-symbolical representation.

2.1 Feature Approach

Objects can be described by a set of discrete features, such as red, round and sweet [Tversky, 1977; McClelland and Rumelhart, 1985]. The similarity between them can be defined as a function of the features they have in common [Osherson, 1995; Sun, 1995; Goldstone, 1999; Gilovich, 1999]. The contrast model of Tversky [Tversky, 1977] is one well-known model in cognitive psychology [Smith, 1995; Opwis and Plötzner, 1996] which describes the similarity between two objects which are described by their features. An object is judged to belong to a verbal category to the extent that its features are predicted by the verbal category [Osherson, 1987]. The similarity of a category C and of a feature set F is given by the following formula, which is inspired by the contrast model of Tversky [Tversky, 1977; Smith, 1995; Opwis and Plötzner, 1996],

$$Sim(C,F) = \frac{|C \cap F|}{|C|} \quad \in [0,1] \tag{1}$$

$|C|$ is the number of the prototypical features that define the category a. The present features are counted and normalized so that the value can be compared. This is a very simple form of representation. A binary vector in which the positions represent different features can represent the set of features. For each category a binary vector can be defined. Overlaps between stored patterns correspond to overlaps between categories.

2.2 The Lernmatrix

The Lernmatrix, also simply called "associative memory" was developed by Steinbuch in 1958 as a biologically inspired model from the effort to explain the psychological phenomenon of conditioning [Steinbuch, 1961; 1971]. Later this model was studied under biological and mathematical aspects by Willshaw [Willshaw *et al.*, 1969] and G. Palm [Palm, 1982; 1990].

Associative memory is composed of a cluster of units. Each unit represents a simple model of a real biological neuron. The Lernmatrix was invented in by Steinbuch, whose goal was to produce a network that could use a binary version of Hebbian learning to form associations between pairs of binary vectors, for example each one representing a cognitive entity. Each unit is composed of binary weights, which correspond to the synapses and dendrites in a real neuron. They are described by $w_{ij} \in \{0,1\}$ in Figure 1. T is the threshold of the unit. We call the Lernmatrix simply *associative memory* if no confusion with other models is possible [Anderson, 1995a; Ballard, 1997].

The patterns, which are stored in the Lernmatrix, are represented by binary vectors. The presence of a feature is indicated by a 'one' component of the vector, its absence through a 'zero' component of the vector. A pair of these vectors is associated and this process of association is called learning. The first of the two vectors is called the *question vector* and the second, the *answer vector*. After learning, the question vector is presented to the associative memory and the answer vector is determined by the retrieval rule.

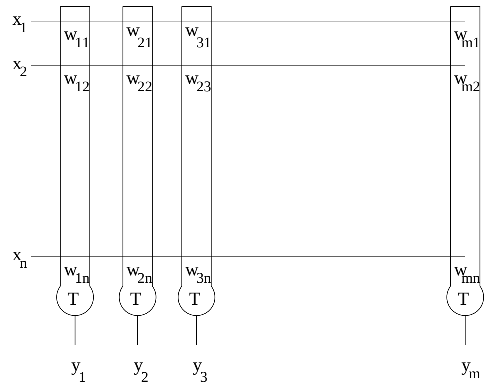

Figure 1: The Lernmatrix is composed of a set of units which represent a simple model of a real biological neuron. The unit is composed of weights, which correspond to the synapses and dendrites in the real neuron. In this figure they are described by $w_{ij} \in \{0,1\}$ where $1 \leq i \leq m$ and $1 \leq j \leq n$. T is the threshold of the unit.

Learning In the initialization phase of the associative memory, no information is stored. Because the information is represented in weights, they are all initially set to zero. In the learning phase, pairs of binary vector are associated. Let $\vec{x}$ be the question vector and $\vec{y}$ the answer vector, the learning rule is:

$$w_{ij}^{new} \begin{cases} 1 & if \;\; y_i \cdot x_j = 1 \\ w_{ij}^{old} & \text{otherwise.} \end{cases} \tag{2}$$

This rule is called the binary Hebbian rule [Palm, 1982]. Every time a pair of binary vectors is stored, this rule is used.

Retrieval In the *one-step* retrieval phase of the associative memory, a fault tolerant answering mechanism recalls the appropriate answer vector for a question vector $\vec{x}$. For the presented question vector $\vec{x}$, the most similar learned $\vec{x^l}$ question vector regarding the Hamming distance is determined and the appropriate answer vector $\vec{y}$ is identified. For the retrieval rule, the knowledge about the correlation of the components is sufficient. The retrieval rule for the determination of the answer vector $\vec{y}$ is:

$$y_i = \begin{cases} 1 & \sum_{j=1}^{n} w_{ij} x_j \geq T \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

where T is the threshold of the unit. The threshold is set as proposed by [Palm *et al.*, 1997] to the maximum of the sums $\sum_{j=1}^{n} w_{ij} x_j$:

$$T := \max_{1 \leq i \leq m} \left\{ \sum_{j=1}^{n} w_{ij} x_j \right\}. \tag{4}$$

Only the units that are maximal correlated with the question vector are set to one.

Storage capacity For an estimation of the asymptotic number of vectorpairs $(\vec{x}, \vec{y})$ which can be stored in an associative memory before it begins to make mistakes in retrieval phase, it is assumed that both vectors have the same dimension n. It is also assumed that both vectors are composed of M 1s, which are likely to be in any coordinate of the vector. In this case it was shown [Palm, 1982; Hecht-Nielsen, 1989; Sommer, 1993] that the optimum value for M is approximately

$$M \doteq \log_2(n/4) \tag{5}$$

and that approximately [Palm, 1982; Hecht-Nielsen, 1989]

$$L \doteq (\ln 2)(n^2/M^2) \tag{6}$$

of vector pairs can be stored in the associative memory. This value is much greater then n if the optimal value for M is used. In this case, the asymptotic storage capacity of the Lernmatrix model is far better than those of other associative memory models, namely 69.31%. This capacity can be reached with the use of sparse coding, which is produced when very small number of 1s is equally distributed over the coordinates of the vectors [Palm, 1982; Stellmann, 1992]. For example an optimal code is defined as following; in the vector of the dimension n=1000000 M=18 ones should be used to code a pattern. The real storage capacity value is lower when patterns are used which are not sparse or are strongly correlated to other stored patterns. Usually suboptimal sparse codes a sufficiently good to be used with the associative memory. An example of a suboptimal sparse code is the representation of words by context-sensitive letter units [Wickelgren, 1969; 1977; Rumelhart and McClelland, 1986; Bentz *et al.*, 1989]. The ideas for the used robust mechanism come from psychology and biology [Wickelgren, 1969; 1977; Rumelhart and McClelland, 1986; Bentz *et al.*, 1989]. Each letter in a word is represented as a triple, which consists of the letter itself, its predecessor, and its successor. For example, six context-sensitive letters encode the word *desert*, namely: *_de, des, ese, ser, ert, rt_*. The character "_" marks the word beginning and ending. Because the alphabet is composed of 26+1 characters, 27^3 different context-sensitive letters exist. In the 27^3 dimensional binary vector each position corresponds to a possible context-sensitive letter, and a word is represented by indication of the actually present context-sensitive letters. We demonstrate the principle of sparse coding by an example of the visual system and visual scene representation.

2.3 Sparse features

In hierarchical models of the visual system [Riesenhuber and Poggio, 1999],[Fukushima, 1980], [Fukushima, 1989], [Cardoso and Wichert, 2010] the neural units have a local view unlike the common fully-connected networks. The receptive fields of each neuron describe this local view. During the categorization the network gradually reduces the information from the input layer through the output layer. Integrating local features into more global features does this. Supposed in the lower layer tow cells recognize two categories at neighboring position, and these two categories are integrated into a more global category. The first cell is named α the second β. The numerical code for α and β may represent the position of each cell. A simple code would indicate if a cell is active or not. One indicates active, zero not active. For c cells we could indicate this information by a binary vector of dimension c. For an image of size $x \times y$ a cell covers the image X times. A binary vector that describes that image using the cell representation has the dimension $c \times X$. For example gray images of the size 128×96 resulting in vectors of dimension 12288 can be covered with:

- 3072 masks M of the size of a size 2×2 resulting in a binary vector that describes that image has the dimension $c_1 \times 3072$, $X_1 = 3072$ (see Figure 2 (a)).
- 768 masks M of the size of a size 4×4 resulting in a binary vector that describes that image has the dimension $c_2 \times 768$, $X_2 = 768$ (see Figure 2 (b)).
- 192 masks M of the size of a size 8×8 resulting in a binary vector that describes that image has the dimension $c_3 \times 192$, $X_3 = 192$ (see Figure 3 (a)).
- 48 masks M of the size of a size 16×16 resulting in a binary vector that describes that image has the dimension $c_4 \times 48$, $X_4 = 48$ (see Figure 3 (b)).

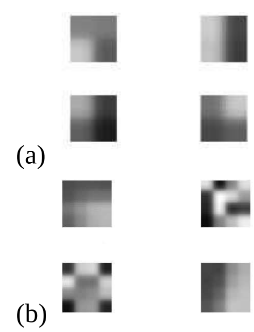

Figure 2: (a) Two examples of of squared masks M of a size 2×2. (b) Two examples of squared masks M of a size 4×4. The masks were learned using simple k-means clustering algorithm.

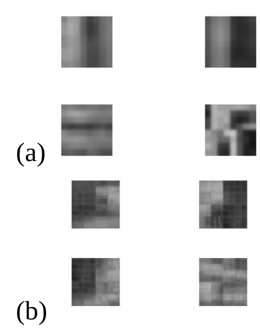

Figure 3: (a) Two examples of of squared masks M of a size 8×8. (b) Two examples of squared masks M of a size 16×16. The masks were learned using simple k-means clustering algorithm.

The ideal c value for a sparse code is related to $M \doteq \log_2(n/4)$.

$$X = \log_2(X \cdot c/4)$$
$$2^X = X \cdot c/4$$
$$c = \frac{4 \cdot 2^X}{X} \quad (7)$$

The ideal value for c grows exponentially in relation to X. Usually the used value for c is much lower then the ideal value resulting in a suboptimal sparse code. The representation of images by masks results in a suboptimal code. The optimal code is approached with the size of masks, the bigger the mask, the smaller the value of X. The number of pixels inside a mask grows quadratic. A bigger masks implies the ability to represent more distinct categories, which implies a bigger c.

An ideal value for c is possible, if the value for $X << 100$. Instead of covering an image by masks, we indicate the present objects. Objects and their position in the visual field can represent a visual scene. A sub-vector of the vector representing the visual scene represents each object. For example, if there is a total of 10 objects, the c value is 409. To represent 409 different categories of objects at different positions resulting in 4090 dimensional binary vector. This vector could represent $\frac{409!}{(409-20)!}$ different visual states of the world. The storage capacity of the associative memory in this case would be around 159500 patterns, which is 28 times bigger as the number of the units (4090).

2.4 Problem Solving

Human problem solving can be described by a problem-behavior graph constructed from a protocol of the person talking aloud, mentioning considered moves and aspects of the situation. According to the resulting theory, searching whose state includes the initial situation and the desired situation in a problem space [Newell, 1990; **?**] solves problems. This process can be described by the production system theory. The production system in the context of classical Artificial Intelligence and Cognitive Psychology is one of the most successful computer models of human problem solving. The production system theory describes how to form a sequence of actions, which lead to a goal, and offers a computational theory of how humans solve problems [Anderson, 1995b]. Production systems are composed of if-then rules that are also called productions. A rule [contains several if patterns and one or more then patterns. A pattern in the context of rules is an individual predicate, which can be negated together with arguments. A rule can establish a new assertion by the then part (its conclusion) whenever the if part (its premise) is true. One of the best-known cognitive models, based on the production system, is SOAR. The SOAR state, operator and result model was developed to explain human problem-solving behavior [Newell, 1990]. It is a hierarchical production system in which the conflict-resolution strategy is treated as another problem to be solved. All satisfied instances of rules are executed in parallel in a temporary mode. After the temporary execution, the best rule is chosen to take action. The decision takes place in the context of a stack of earlier decisions. Those decisions are rated utilizing preferences and added to the stack by chosen rules. Preferences are determined together with the rules by an observer using knowledge about a problem.

According to the production system theory, we can define a geometrically based problem-solving model as a production system operating on vectors of fixed dimensions. Instead of rules, we use associations and vectors represent the states. Our goal is to form a sequence of associations, which lead to a desired state represented by a vector, from an initial state represented by a vector. Each association changes some parts of the vector. In each state, several possible associations can be executed, but only one has to be chosen. Otherwise, conflicts in the representation of the state would occur. To perform these operations, we divided a vector representing a state into sub-vectors. An association recognizes some sub-vectors in the vector and exchanges them for different sub-vectors. It is composed of a precondition of fixed arranged m sub-vectors and a conclusion. Suppose a vector is divided into n sub-vectors with $n > m$. An association recognizes m different sub-vectors and exchanges them for different m sub-vectors. To recognize m sub-vectors out of n sub-vectors we perform a permutation p(n,m) and verify if each permutation corresponds to a valid precondition of an association. For example, if there is a total of 7 elements and we are selecting a sequence of three elements from this set, then the first selection is one from 7 elements, the next one from the remaining 6, and finally from the remaining 5, resulting in 7 * 6 * 5 = 210, see Figure 4.

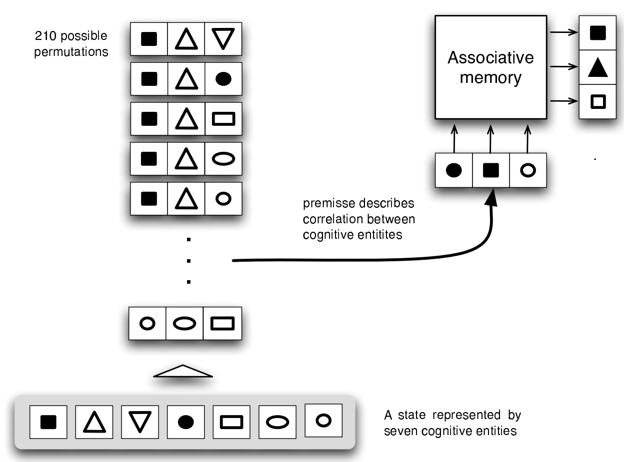

Figure 4: To recognize one learned association permutations are formed. For example, if there is a total of 7 elements and we are selecting a sequence of three elements from this set, then the first selection is one from 7 elements, the next one from the remaining 6, and finally from the remaining 5, resulting in 7 * 6 * 5 = 210. In our example, all possible three-permutations sub-vectors of seven sub-vectors are formed to test if the precondition of an association is valid.

Out of several possible associations, we chose the one,

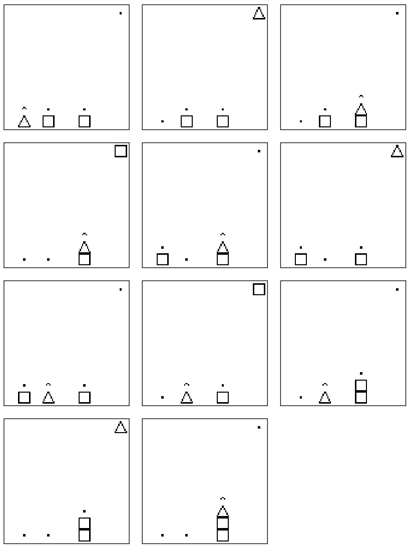

Figure 5: The simplest method corresponds to a random choice, and does not offer any advantage over simple symbolical representation. An example of visual planning of the tower building task of three blocks using the random choice is shown. The upper left pattern represents the initial state; the bottom right pattern, the desired state.

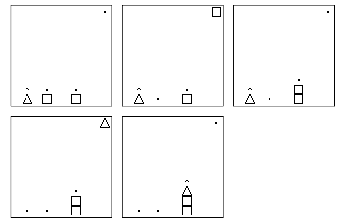

Figure 6: The computation can be improved by a simple and universal heuristics function, which takes into account the relationship between the vector and the corresponding similarity of the represented object or states in the real world as expressed by Equation 1 for binary vectors. The heuristics function makes a simple assumption that the distance between the states in the problem space is related to the distance between the sub-symbols representing the visual states. The distance between the states in the problem space is related to the distance between the visual state. An example of visual planning of the tower building task of three blocks using hill climbing using the similarity function, see Equation 1. The upper left pattern represents the initial state; the bottom right pattern, the desired state.

which modifies the state in such a way that it becomes more similar to the desired state according to the Equation 1. The desired state corresponds to the category of Equation 1, each feature represents a possible state. The states are represented by sparse features. With the aid of this heuristic hill climbing is performed. Each element represents an object. Objects are represented by some dimensions of the space and form a sub-space by themselves, see Figure 4.

The computation can be improved by a simple and universal heuristics function, which takes into account the relationship between the vector and the corresponding similarity of the represented states see Figure 5 and Figure 6. The heuristics function makes a simple assumption that the distance between the states in the problem space is related to the similarity of the vectors representing the states.

The similarity between the corresponding vectors can indicate the distance between the sub-symbols representing the state. Empirical experiments in popular problem-solving domains of Artificial Intelligence, like robot in a maze, block world or 8-puzzle indicated that the distance between the states in the problem space is actually related to the similarity between the images representing the states [Wichert, 2001; Wichert *et al.*, 2008; Wichert, 2009].

3 Conclusion

Living organisms experience the world as a simple. The actual perception of the world and manipulation in the world by living organisms lead to the invention or recreation of an experience that, at least in some respects, resembles the experience of actually perceiving and manipulating objects in the absence of direct sensory stimulation. This kind of representation is called sub-symbolic. Sub-symbolic representation implies heuristic functions. The assumption that the distance between states in the problem space is related to the similarity between the sub-symbols representing the states is only valid in simple cases. However, simple cases represent the majority of exiting problems in domain. Sense organs sense the world by receptors which a part of the sensory system and the nervous system. Sparse binary vectors can describe sub-symbolic representation, which can be efficiently stored in associative memories. A simple code would indicate if a receptor is active or not. One indicates active, zero not active. For c receptors we could indicate this information by a binary vector of dimension c with only one ”1”, the bigger the c, the sparser the code. For receptors in X positions the sparse code results in $c \times X$ dimensional vector with X ones.

Acknowledgments

This work was supported by Fundao para a Cencia e Tecnologia (FCT) (INESC-ID multiannual funding) through the PIDDAC Program funds.

References

[Anderson, 1995a] James A. Anderson. *An Introduction to Neural Networks*. The MIT Press, 1995.

[Anderson, 1995b] John R. Anderson. *Cognitive Psychology and its Implications*. W. H. Freeman and Company, fourth edition, 1995.

[Ballard, 1997] Dana H. Ballard. *An Introduction to Natural Computation*. The MIT Press, 1997.

[Bentz *et al.*, 1989] Hans J. Bentz, Michael Hagstroem, and Günther Palm. Information storage and effective data retrieval in sparse matrices. *Neural Networks*, 2(4):289–293, 1989.

[Cardoso and Wichert, 2010] A. Cardoso and A Wichert. Neocognitron and the map transformation cascade. *Neural Networks*, 23(1):74–88, 2010.

[Fukushima, 1980] K. Fukushima. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern*, 36(4):193–202, 1980.

[Fukushima, 1989] K. Fukushima. Analisys of the process of visual pattern recognition by the neocognitron. *Neural Networks*, 2:413–420, 1989.

[Gilovich, 1999] Thomas Gilovich. Tversky. In *The MIT Encyclopedia of the Cognitive Sciences*, pages 849–850. The MIT Press, 1999.

[Goldstone, 1999] Robert Goldstone. Similarity. In *The MIT Encyclopedia of the Cognitive Sciences*, pages 763–765. The MIT Press, 1999.

[Hecht-Nielsen, 1989] Robert Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, 1989.

[Kosslyn, 1994] Stephen M. Kosslyn. *Image and Brain, The Resolution of the Imagery Debate*. The MIT Press, 1994.

[Lancy, 1983] D.F. Lancy. *Cross-Cultural Studies in Cognition and Mathematics*. Academic Press, New York, 1983.

[McClelland and Rumelhart, 1985] J.L. McClelland and D.E. Rumelhart. Distributed memory and the representation of general and specific memory. *Journal of Experimental Psychology: General*, 114:159–188, 1985.

[Newell and Simon, 1976] A. Newell and H. Simon. Computer science as empirical inquiry: symbols and search. *Communication of the ACM*, 19(3):113–126, 1976.

[Newell, 1990] Allen Newell. *Unified Theories of Cognition*. Harvard University Press, 1990.

[Opwis and Plötzner, 1996] Klaus Opwis and Rolf Plötzner. *Kognitive Psychologie mit dem Computer*. Spektrum Akademischer Verlag, Heidelberg Berlin Oxford, 1996.

[Osherson, 1987] Daniel N. Osherson. New axioms for the contrast model of similarity. *Journal of Mathematical Psychology*, 31:93–103, 1987.

[Osherson, 1995] Daniel N. Osherson. Probability judgment. In Edward E. Smith and Daniel N. Osherson, editors, *Thinking*, volume 3, chapter two, pages 35–75. MIT Press, second edition, 1995.

[Palm *et al.*, 1997] G. Palm, F. Schwenker, F.T. Sommer, and A. Strey. Neural associative memories. In A. Krikelis and C. Weems, editors, *Associative Processing and Processors*, pages 307–325. IEEE Press, 1997.

[Palm, 1982] Günther Palm. *Neural Assemblies, an Alternative Approach to Artificial Intelligence*. Springer-Verlag, 1982.

[Palm, 1990] Günther Palm. Assoziatives Gedächtnis und Gehirntheorie. In *Gehirn und Kognition*, pages 164–174. Spektrum der Wissenschaft, 1990.

[Riesenhuber and Poggio, 1999] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999.

[Rumelhart and McClelland, 1986] D.E. Rumelhart and McClelland. On learning the past tense of english verbs. In J.L. McClelland and D.E. Rumelhart, editors, *Parallel Distributed Processing*, pages 216–271. MIT Press, 1986.

[Simon, 1991] Herbert A. Simon. *Models of my Life*. Basic Books, New York, 1991.

[Smith, 1995] Edward E. Smith. Concepts and categorization. In Edward E. Smith and Daniel N. Osherson, editors, *Thinking*, volume 3, chapter one, pages 3–33. MIT Press, second edition, 1995.

[Sommer, 1993] Friedrich T. Sommer. *Theorie neuronaler Assoziativspeicher*. PhD thesis, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, 1993.

[Steinbuch, 1961] K. Steinbuch. Die Lernmatrix. *Kybernetik*, 1:36–45, 1961.

[Steinbuch, 1971] Karl Steinbuch. *Automat und Mensch*. Springer-Verlag, fourth edition, 1971.

[Stellmann, 1992] Uli Stellmann. *Ähnlichkeitserhaltende Codierung*. PhD thesis, Universität Ulm, Ulm, 1992.

[Sun, 1995] Ron Sun. A two-level hybrid architecture for structuring knowledge for commonsense reasoning. In Ron Sun and Lawrence A. Bookman, editors, *Computational Architectures Integrating Neural and Symbolic Processing*, chapter 8, pages 247–182. Kluwer Academic Publishers, 1995.

[Tarski, 1956] Alfred Tarski. *Logic, Semantics,Metamathematics*. Oxford University Press, London, 1956.

[Tversky, 1977] A. Tversky. Feature of similarity. *Psychological Review*, 84:327–352, 1977.

[Wichert *et al.*, 2008] A. Wichert, J. D. Pereira, and P. Carreira. Visual search light model for mental problem solving. *Neurocomputing*, 71(13-15):2806–2822, 2008.

[Wichert, 2001] Andrzej Wichert. Pictorial reasoning with cell assemblies. *Connection Science*, 13(1), 2001.

[Wichert, 2009] Andreas Wichert. Sub-symbols and icons. *Cognitive Computation*, 1(4):342–347, 2009.

[Wickelgren, 1969] Wayne A. Wickelgren. Context-sensitive coding, associative memory, and serial order in (speech)behavior. *Psychological Review*, 76:1–15, 1969.

[Wickelgren, 1977] Wayne A. Wickelgren. *Cognitive Psychology*. Prentice-Hall, 1977.

[Willshaw *et al.*, 1969] D.J. Willshaw, O.P. Buneman, and H.C. Longuet-Higgins. Nonholgraphic associative memory. *Nature*, 222:960–962, 1969.