Neural associative memories and sparse coding

Günther Palm

University of Ulm, Institute of Neural Information Processing, D-89069 Ulm, Germany

ARTICLE INFO

Keywords:
Associative memory
Sparse coding
Cortical networks

ABSTRACT

The theoretical, practical and technical development of neural associative memories during the last 40 years is described. The importance of sparse coding of associative memory patterns is pointed out. The use of associative memory networks for large scale brain modeling is also mentioned.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Associative memory has been an active topic of research for more than 50 years and is still investigated both in neuroscience and in artificial neural networks. The workings of associations in human memory have probably first been addressed in psychology and even philosophy (David Hume has already stated rules of association).

The basic observation of association occurs when we try to find a specific piece of information in our memory and we do not retrieve it immediately. In such cases we notice that the present state of our mind or brain which presumably contains aspects of the present situation and contextual information pointing at the missing piece (momentarily however not sufficient to find it), starts a sequential process of associations from one item to the next (possibly governed by semantic similarity) that eventually ends in the present state.

From a technical point of view there are two different mechanisms that are needed in this process of association: hetero-association, that leads from one pattern to the next, and auto-association from one pattern to itself, that is useful for the recognition of one pattern as best fitting, or also for slight correction or completion of this pattern, and thereby ending the chain of (hetero-) associations. There is another technical type of associative memory that is often mentioned (and, in principle, could be regarded as a special case of auto-association), namely bidirectional association that goes back and forth between two patterns A and B. Simple graphical representations of these three types of associative memories are shown in Fig. 2.

Neuroscientists are typically not content with a mere phenomenological description of the process of association in the mind on a cognitive level, they want to relate it to concrete neuro-physiological mechanisms in the brain. The first concrete hypothesis in this direction goes back to the psychologist Donald Hebb (1949) who formulated a rule for synaptic plasticity that postulates an increase in synaptic connection strength induced by coincident activity in the two neurons connected by the synapse. This idea has led to a lot of very fruitful experimental investigations that eventually confirmed Hebb’s ideas (see Caporale & Dan, 2008 for a recent review).

On the technical side this has led to the development of Neural Associative Memory (NAM) models based on matrix calculus, where a memory storage matrix \( C = (c_{ij}) \) is formed that contains the weights \( c_{ij} \) of synaptic connections between neurons \( i \) and \( j \). In hetero-association these connections connect an input pool of neurons (containing neuron \( i \)) to an output pool of neurons (containing neuron \( j \)). In auto-association they typically connect one pool of neurons back to itself in a recurrent fashion. The roots for this “basic NAM formalism” can be found in engineering (Steinbuch, 1961), in particular in holographic memory implementations (Gabor, 1969; Longuet-Higgins, Willshaw, & Buneman, 1970), and in early neural network modeling (e.g. Amari, 1972; Anderson, 1968; Anderson, Silverstein, Ritz, & Jones, 1977; Dunin-Barkowski & Larionova, 1985; Little, 1974; Marr, 1969; Wigström, 1975). The first systematic overviews were given by Kohonen (1977) and Palm (1980).

2. Basic NAM formalism

Given a set of (pairs of) patterns \((x^A, y^B)\) to be stored in the matrix \( C \), the process of storage (matrix formation (1)) and retrieval (activity propagation (2)) can usually be described by the following equations:

\[
\begin{align*}
    c_{ij} &= \sum_{\mu} x^A_{i\mu} y^B_{j\mu} \quad \text{(additive rule)} \\
    &\quad \text{or} \quad c_{ij} = \max_{\mu} x^A_{i\mu} y^B_{j\mu} \quad \text{(binary rule)} \\
    u &= xC \quad \text{and} \quad y_j = \begin{cases} 
        1 & \text{if } u_j \geq \theta \\
        0 & \text{otherwise.}
    \end{cases}
\end{align*}
\]
Here $\theta$ is an appropriately chosen threshold.

Considering a sequential memory storage process it is natural to describe the formation of the matrix $C$ as a sequence of memory or learning steps in which one more pair $(x^i, y^i)$ is added to the memory. In each learning step (at time $t$) the change $\Delta c_{ij}$ of the entry $c_{ij}$ of the matrix $C$ depends only on the product $x_i y_j$ of the $i$-th coordinate of the input $x$ and the $j$-th coordinate of the output $y$, i.e. only on the presynaptic activity $x_i$ and the postsynaptic activity $y_j$ at the synapse modeled by $c_{ij}$ at time $t$. Thus the synaptic changes can be computed locally in space and time, and Eq. (1) is called a local learning rule (Palm, 1982, 1992). Local learning rules are biologically plausible and computationally simple; in particular they are useful for parallel implementation.

In NAM systems the output of the neurons is usually considered as binary. In the past either $\{-1, 1\}$ or $\{0, 1\}$ have been used as the binary values of the stored and retrieved patterns. The use of $\{-1, 1\}$ goes back to John Hopfield (1982). His version of an additive NAM, the Hopfield memory model, has turned out to be quite inefficient as a memory. This is due to two factors:

1. The “Hopfield learning rule” $\Delta c_{ij} = x_i y_j$ for $x_i, y_j \in \{-1, 1\}$ changes every entry $c_{ij}$ of the matrix $C$ in every learning step.
2. The changes go in both directions (up and down), so they can cancel each other.

This is actually quite different for the “Hebb learning rule” $\Delta c_{ij} = x_i y_j$ for $x_i, y_j \in \{0, 1\}$ which has been considered in earlier investigations of NAM going back to Steinbuch (1961) with a first asymptotic analysis given by Willshaw, Buneman, and Longuet-Higgins (1969), see also Willshaw (1971).

Towards the end of the 1980s it became clear that $\{0, 1\}$ and the corresponding Hebb learning rule should be used in practical applications and that sparseness of the stored patterns is most important for an effective use of NAMs for information storage and retrieval (Palm, 1988, 1990; Tsodyks & Feigelman, 1988). The importance of sparseness was already implicit in the early analysis of Willshaw (1971), but it was only made explicit a few years later by myself (Palm, 1980, 1985, 1987). Sparseness is the basis for the efficiency of technical applications and VLSI realizations of NAMs (see Palm & Bonhoeffer, 1984, US Patent No. 4777622 (1988) and Palm & Palm, 1991).

3. Information capacity and critical capacity

In order to demonstrate the importance of sparseness in associative memory patterns and to prove the efficiency of sparse NAMs it was necessary to develop a clean definition of the information capacity of NAMs. Such a definition is best given in terms of information theory, considering the total amount of information that can effectively be stored in and retrieved from an associative memory matrix of a given size (Palm, 1980, 1992; Palm & Sommer, 1992, 1995). Using proper definitions it was possible to show that an asymptotical (large memory matrices) optimal capacity of $2^{0.69} \approx 0.72$ bit per matrix entry can be achieved for sparse memory patterns with the binary storage version (Palm, 1980), and $1/(2 \ln 2) \approx 0.72$ bit per matrix entry can be achieved with the additive storage version (Palm, 1988, 1990). The difference between these two values is quite small; in the binary version, however, one clearly needs just one hardware bit for one matrix entry, whereas one needs somewhat more hardware bits per entry in the additive version. These results were actually calculated for hetero-association; they also hold for bidirectional association (Sommer & Palm, 1999); for auto-association the information capacity is just half as large, corresponding also to the symmetry of the memory matrix $C$. Many more results concerning information capacity have been summarized in my earlier review article (Palm, 1991).

Initiated by the paper of Hopfield (1982) many theoretical physicists became interested in associative memory and applied methods from the theory of spin-glasses to the analysis of feedback auto-associative memories (Fig. 2(b) or (c)) as dynamical systems with a nice natural energy function

$$H(x) = -x C x^T$$

governing the asymptotic behavior, resulting (for symmetric $C$) in an attractor dynamics towards the minima of $H(x)$ as fixed points (Amit, 1989; Amit, Gutfreund, & Sompolinsky, 1987; Domany, van Hemmen, & Schulten, 1991; Hopfield, 1982). Concerning the use of these systems as practical associative memories, the most important questions are

1. Can we construct the matrix $C$ in such a way that a prespecified set $M$ of memory vectors becomes fixed points?
2. How large is $M$ as compared to the matrix dimension or network size $n$ (the critical capacity $\alpha = M/n$)?
3. How large are the “basins of attraction”, i.e. how many errors in an input pattern can be corrected by the feedback retrieval dynamics?

The first two questions were studied extensively. In a nutshell the two most important results are that $\alpha = 2$ can be reached asymptotically in principle (Gardner, 1987, 1988), but there is no local rule to construct the appropriate matrix $C$ from the memory set $M$ and the entries of the matrix $C$ need to be stored with high accuracy. And secondly, $\alpha \approx 0.14$ can be reached with the additive Hopfield rule (Eq. (1)) (Amit et al., 1987; Hopfield, 1982). Also the binary (or “clipped”) storage version has been considered, leading to considerably lower values for $\alpha$ (e.g. $\alpha \approx 0.83$ for the non-constructive binary case was found by Krauth & Mézard, 1989).

The third question has also been studied (e.g. Horner, Bormann, Frick, Kinzelbach, & Schmidt, 1989; Nadal & Toulouse, 1990; Opper, Kleinz, Kohler, & Kinzel, 1989). However, satisfactory answers were at first only found for the non-local construction of $C$ and only for $\alpha$-values that are considerably lower than the critical capacity (for the binary memory matrix reasonable error correction is possible for $\alpha$-values up to about 0.3 Opper et al., 1989), allowing a possible correction of about 4% of the entries. In this case ($\alpha = 0.3$) the information capacity is about 0.06. Numerical simulations of the Hopfield memory show that the basins of attraction are surprisingly small, corresponding to an information capacity of less than 0.04. Perhaps the most important reason for this is the large number of spurious (i.e. unwanted) fixed points that are created by the Hopfield rule. Their number seems to increase exponentially with the size of $M$.

From the application perspective the results for the Hopfield rule are far below the results achieved with the Hebb rule (e.g. $0.14$ vs. $0.69$). On top of that, there are two additional reasons why the information capacity (total information divided by $n^2$) is usually considerably smaller than the critical capacity of the same memory: First the information that can be extracted from a fixed point is always less and usually much less than $n$ bits. Thus the information capacity will be much less than the critical capacity. Secondly, for sparse binary patterns with a probability $p$ for a 1-entry, the total information content of one pattern is roughly $-np \log p$ which is again less than $n$. The second effect has been incorporated into the definition of the critical capacity $\alpha$ for sparse (or biased) memory patterns (e.g. Gardner, 1987).
The main reason for the large difference in performance is the sparseness of the memory patterns used with the Hebb rule in the so-called Willshaw model (Willshaw et al., 1969). This parameter range has not been well-treated in the early physics literature, probably due to the misleading symmetry assumption (symmetry with respect to sign change) that was imported from spin-glass physics. This prevented the use of binary $[0,1]$ activity values and the corresponding Hebb rule and the discovery of sparseness. It also led to unrealistic neural models, both concerning neural activity (an active neuron carries more information than a passive one) and connectivity (a synaptic weight cannot cross the boundary between excitatory, positive and inhibitory, negative). Only in 1988, it was the contribution of Mischa Tsodyks that brought $[0,1]$ activity modeling, the Hebb rule for synaptic plasticity and sparseness to a broader recognition in the theoretical physics community (Tsodyks & Feigelman, 1988). He showed that for sparse Hebbian associative memories $\alpha = \frac{1}{2} \ln 2 \approx 0.72$, corresponding to an information capacity (for auto-associative pattern completion) of about 0.18 (Palm, 1988, 1991; see also Schwenker, Sommer, & Palm, 1996). The corresponding older result for the sparse binary Willshaw model is $\alpha = \ln 2$ resulting in an information capacity of $\frac{1}{2} \ln 2 \approx 0.17$ (Palm, 1991; see also Palm & Sommer, 1992; Schwenker et al., 1996).

4. Sparse coding

In technical applications of NAM, efficiency is clearly an important issue. This involves not only storage capacity or storage efficiency, but also fast retrieval of the stored patterns. Since retrieval is simply done by vector–matrix multiplication with entries in $[0, 1]$, this is reduced to counting, followed by thresholding, so it is very fast. If the input patterns are sparse it is even faster, since the number of operations is simply proportional to the number of 1-entries in an input pattern. So it is practically important to use sparse binary patterns. This raises the problem of sparse coding. At first sight this appears as no big problem. If one wants a sparse representation for a fixed number $M$ of items, for example in terms of binary vectors that each contain $k$ 1-entries and $n-k$ 0-entries, then there are $\binom{n}{k}$ such vectors and one can easily map the $M$ items into such patterns if $M \leq \binom{n}{k}$. Another possibility is to use a concatenation of several 1-out-of-$n$ codes to create a k-out-of-kn code (if $M \leq n^k$).

However, if one wants to make use of the nice property of NAMs that they respect pattern similarity (in the sense of overlap, inner product, or Hamming distance of binary patterns), then one has to represent semantically similar items by similar binary vectors (Baum, Moody, & Wilczek, 1988; Palm, Schwenker, & Sommer, 1994; Palm, Schwenker, Sommer, & Strey, 1997). This problem of similarity based sparse coding has already been treated systematically by Stellmann (1992) by investigating methods that can generate roughly similarity preserving sparse binary code vectors from a given similarity matrix. Also, in many practical applications there is a natural way of generating sparse code vectors: In many pattern recognition or classification tasks with a moderately large number of classes (e.g. in spoken word recognition, face recognition, written letter recognition) it is usual practice to output a 1-out-of-$n$ binary vector representation of the $n$ classes. In more complex applications with very many classes to be distinguished one often uses a more structured approach that describes each class by a large number of binary features, which often are sparse again. If these binary-feature-based representations are not sparse enough, it often makes sense to combine two or more of those features into one (creating a 1-out-of-kn representation from a 1-out-of-k and a 1-out-of-n representation). Of course, this does not make sense (in terms of similarity) for any arbitrary combination of features. These more practical issues of creating sparse codes with natural similarity have recently been rediscovered in practical applications such as visual object recognition (Ganguli & Sompolsky, 2012; Kavukcuoglu, Lecun, & LeCun, 2010; Kreutz-Delgado et al., 2003; Lee, Battle, Raina, & Ng, 2007; Szlam, Gregor, & LeCun, 2012).

Even on the level of sensor outputs, signals are often sparse, because only changes of the output are reported as temporally separated events. Of course, this strongly depends on the type of sensor. In video signals, for example, the most common compression codes are essentially based on the sparseness of signal differences, both in time and visual (2d) space. The same principle is also working in the human or animal visual system resulting in center-surround antagonistic activation of retinal ganglion cells and sparse activity of edge-detecting cells in the primary visual cortex (Field, 1987; Olshausen, 2003b; Olshausen & Field, 1996a, 1996b; Vinje & Gallant, 2000).

5. The sparseness principle

Also outside the context of associative memory sparseness seems to be a useful principle in machine learning and signal processing (Candes & Romberg, 2007; Coulter, Hillar, Isley, & Sommer, 2010; Donoho & Elad, 2003; Hillar & Sommer, 2011; Hoyer, 2004; Hoyer & Hyvärinen, 2002; Hurley & Rickard, 2009; Kavukcuoglu et al., 2010; Kreutz-Delgado et al., 2003; Szlam et al., 2012), so that one can often expect sparse representations as a result of this kind of processing. In machine learning, in particular in unsupervised or semisupervised learning one tries to create useful compact representations of the data to be learned by autoencoding networks or component analysis (PCA, ICA). In this context one often uses techniques of regularization to obtain robust representations and avoid overfitting. Here again sparseness constraints have turned out to be extremely useful, leading to overcomplete sparse representations. The reasons for this common observation are currently not yet well understood although sparse sensory representations have been investigated since the late 1990s, for example by Bruno Olshausen and others (e.g. Carlson, Rasquinha, Zhang, & Connor, 2011; Coulter et al., 2010; Földiák & Young, 1998; Ganguli & Sompolsky, 2012;
Also in sensor fusion sparse representations (often coming from these sources) can be very useful, but here one can perhaps give a hint of the reason. When we want to fuse two sparse binary representations $x$ and $y$ of two types of sensor data coming from the same object, we can learn to associate the positive (nonzero) features in $y$ with those in $x$. If these features are sparse, then the co-occurrence of a feature in $x$ with a feature in $y$ is much more significant, i.e. much more unlikely to happen by chance, compared to the non-sparse case. Thus we are learning or associating less, but more significant, events, which is likely to result in a better performance.

In the neurosciences today it is commonplace that spiking neural activity is mostly sparse (e.g. Carlson et al., 2011; Földiák & Young, 1998; Franco, Rolls, Aggelopoulos, & Jerez, 2007; Hahnloser, Kozhevnikov, & Fee, 2002; Koulakov & Rinberg, 2011; Vinje & Gallant, 2000; Wolfe, Houweling, & Brecht, 2010). Considering spike-trains of single neurons the argument is simply that the duration of a spike is typically less than a millisecond, which would allow for up to 1000 spikes per second, whereas the observed spike frequencies are much lower; even a very active neuron hardly produces more than 100 spikes per second. A more qualitative observation is that observed spike frequencies tend to go down when we move from sensory or motor systems to more central brain regions like the cerebral cortex. It is not easy to say what the average spike frequency of a cortical neuron may be during a typical day, but it is most likely not more than about 5 spikes per second. This observed sparsity of neural spiking may of course be due to an energy saving principle (Lennie, 2003), but it may also be related to the working of associative synaptic plasticity in the cortex, i.e. to storage efficiency. Concerning the functional role of sparse activity in the cortex, there are several theoretical ideas, in particular for the learning and generation of sparse representations (e.g. Földiák, 1990; Hyvärinen, 2010; Lee et al., 2007; Olshausen, 2003a; Perrinet, 2010; Rehn & Sommer, 2006, 2007; Rozell, Johnson, Baraniuk, & Olshausen, 2008; Zetzsche, 1990).

Also on a cognitive psychological level sparseness appears to be very natural. Most of our mental concepts (or the words for them) occur sparsely. This becomes immediately obvious when we consider negation. We cannot really imagine something like a non-car or a non-table, because this would encompass essentially everything and cannot be conceived. Thus our usual concepts signify rather small, compact and rare constellations of sensations.

### 6. Technical realization of NAMs

During the 1990s some serious attempts were made to develop technical hardware realizations for massively parallel computing of NAM functionalities (Gao & Hammerstrom, 2003; Heittmann, Malin, Pintaske, & Rückert, 1997; Heittmann & Rückert, 1999, 2002; Zhu & Hammerstrom, 2002). The basic idea is to use a large number of conventional RAM chips storing columns of the storage matrix with parallel counting and thresholding (Palm & Bonhoeffer, 1984; Palm & Palm, 1991).

Another essential idea is to use an address bus to communicate the NAM activity patterns between processors. This is important because generally the inter-process communication is always the bottleneck in massively parallel computing architectures. Here the use of sparse activity patterns makes it possible to save on transmission rate by just transmitting the addresses of the few active neurons in a NAM population. This idea has always been used in our own parallel architectures (e.g. Palm & Palm, 1991; Palm et al., 1997) and later it has been widely adopted in the parallel implementation of spiking or pulse-coded neural networks (e.g. Mahowald, 1994) and even received an acronym, AER, i.e. address-event representation. Now these ideas are used in several larger projects aimed at hardware implementations of large-scale spiking neural networks for technical applications or for brain simulations (De Garis, Shuo, Goertzel, & Ruiting, 2010; Djurfeldt et al., 2008; Fay, Kaufmann, Knoblauch, Markert, & Palm, 2005; Johansson & Lansner, 2007; Markram, 2006; Seiffert, 2004; Zhu & Hammerstrom, 2002).

In such application oriented projects there also occurred the idea of developing efficient software implementations in terms of sparse matrix operations using pointers to the nonzero elements (Bentz, Hagstroem, & Palm, 1989). In these applications there already appeared some indications that the limit of in 2 retrievable bits per hardware bit, i.e. the efficiency limit of in 2 can be surpassed. Only recently we were able to show that this is indeed the case (Knoblauch, 2011; Knoblauch, Palm, & Sommer, 2010). By optimizing the quotient of the information storage capacity and the information contained in the storage matrix itself, we found a regime of ultra-sparse memory patterns, where the storage matrix is also a sparse matrix and the quotient, i.e. the efficiency of the memory approaches 1.

### 7. More detailed modeling of NAMs inspired by neuroscience

Following the early ideas of Donald Hebb (1949), the stepwise formation of an associative memory matrix is understood as a rule for the change of synaptic efficacies in the synaptic connectivity matrix connecting the neurons of the NAM. Thus associative learning is realized by synaptic plasticity, which was a subject of intensive investigation in the neurosciences. In neuroscience and recently also in neuromorphic engineering there has been increasing interest in spiking neuron models and the role of spike synchronization (e.g. Jin, Furber, & Woods, 2008; Knoblauch & Palm, 2001; Mehrtash et al., 2003; Palm & Knoblauch, 2005; Plana et al., 2011; Serrano-Gotarredona et al., 2008). In this context the mechanisms for Hebbian synaptic plasticity has been analyzed and modeled on a finer time-scale as a long-term process of synaptic modification that is triggered by the coincidence or close temporal succession of single pre- and postsynaptic spikes (or sometimes two or three of these spikes). These more detailed versions of the general idea of Hebbian synaptic plasticity were observed in neurophysiology (e.g. Bi & Poo, 1998; Caporale & Dan, 2008; Froemke, Poo, & Dan, 2005; Markram, Lübke, Frotscher, & Sakmann, 1997; Song, Miller, & Abbott, 2000) and are called spike-timing dependent plasticity (STDP).

Their analysis (e.g. Izhikevich & Desai, 2003; Kempter, Gerstner, & Van Hemmen, 1999; Pfister & Gerstner, 2006) has led to some controversial arguments regarding the consistency between the observed requirement of temporal ordering of pre- and postsynaptic spikes (pre- slightly before post-) and the formation and stabilization of recurrent auto-associations (Clopath, Büsing, Vasilaki, & Gerstner, 2010; Knoblauch & Hauser, 2011). In many STDP models exactly synchronized spikes in two neurons result in a weakening of the synapse connecting them. So synchronized spiking activity in a recurrent assembly would destroy the assembly. However, already a slight temporal jittering of these synchronized spikes in the range of milliseconds can reverse this effect and lead to a strengthening of the synapse. Now the current debate is, what is more likely to happen in a biologically realistic parameter range and in neurobiological reality (Knoblauch, Hauser, Gewaltig, Körner, & Palm, 2012). Currently this interesting discussion can certainly not be interpreted as a conclusive falsification of the assembly ideas, it rather shows new technical possibilities, in particular if one moves from a local...
interpretation of assemblies to a more global systemic one, where the network of the whole cortex is considered as a machine for learning and organization of behavior.

In such a more constructive fashion it is easily possible to create larger systems of several cortical modules based on hetero- and auto-associative connectivity structures that work with spiking neuron models producing biologically plausible single neuron and population dynamics and that can interact in a functionally meaningful fashion to generate computationally demanding behavior which may be called cognitive (Fransén & Lansner, 1998; Lansner, 2009; Lansner & Fransén, 1992; Sandberg, Tegner, & Lansner, 2003). We produced an example of such a system (see Fig. 3) that contains about 30 cortical modules or areas and demonstrates the understanding of simple command sentences by controlling a robot to perform the appropriate actions (Fay et al., 2005; Knoblauch, Markert, & Palm, 2005; Markert, Knoblauch, & Palm, 2007). The basic idea of such an approach is to model the cerebral cortex as a network of associative memory modules. I have developed this idea already in my book ‘Neural Assemblies’ (Palm, 1982). It was strongly influenced by intense discussions with Valentino Braitenberg and by his analysis and interpretation of the anatomical cortical connectivity (see Braitenberg, 1977, 1978; Braitenberg & Schüz, 1998). Valentino also had the rather cute idea to codify the basic concept of a recurrent associative memory module (see Fig. 2(e) and Palm, 1980) in a widely visible logo, namely the logo of the Springer book series ‘Studies of Brain Function’ which started in 1977. Incidentally this logo also has some similarity to the logo of ‘Neural Networks’.

8. Conclusion

Theoretically it is no problem to show computational universality of binary or spiking neuron networks. The first results on this topic go back to McCulloch and Pitts’ paper and to early work in computer science, notably by Kleene. Later this topic was taken up again in a wider context by Wolfgang Maass and others (e.g. Funahashi & Nakamura, 1993). The same is of course also true for associative memory networks (Wennekers & Palm, 2007), which may be used for a higher-level psychologically more plausible implementation of thought processes or human problem solving capabilities. This type of higher level brain modeling based on networks of larger modules, each containing several populations of (thousands of) neurons, may eventually bring us closer to the goal of early neuroscientists like Donald Hebb or Warren McCulloch, namely to bridge the huge gap between lower level neuroscientific analysis of brain activity and higher level synthetic psychological descriptions of human cognition, by creating an additional process description (in terms of Hebbian cell assemblies and cortical modules) at an intermediate level that is amenable to interpretations from both sides. In the recent literature one can find several projects or schools that may be associated with such a program (e.g. Edelman & Tononi, 2000, Hawkins & Blakeslee, 2004 and Hecht-Nielsen, 2007 and of course also Steve Grossberg and John Taylor) and most of them are entertaining ideas that are based on, closely related to, or at least easily translatable into associative memory models.

Acknowledgments

I would like to thank Miriam Schmidt and Martin Schels for helping me with the final preparation of the manuscript.

References


