Available online at www.sciencedirect.com
ScienceDirect
Physica D 237 (2008) 1207–1214
PHYSICA D
www.elsevier.com/locate/physd

Analog-symbolic memory that tracks via reconsolidation

Hava T. Siegelmann*

Department of Computer Science, University of Massachusetts, Amherst, MA 01003, USA
Program for Evolutionary Dynamics Harvard University, One Brattle Square, Suite 6, Cambridge, MA 02138-3758, USA

Available online 30 March 2008

Abstract

A fundamental part of a computational system is its memory, which is used to store and retrieve data. Classical computer memories rely on the static approach and are very different from human memories. Neural network memories are based on auto-associative attractor dynamics and thus provide a high level of pattern completion. However, they are not used in general computation since there are practically no algorithms to load an arbitrary landscape of attractors into them. In this sense neural network memory models cannot communicate well with symbolic and prior knowledge.

We propose the design of a new memory based on localist attractor dynamics with reconsolidation called Reconsolidation Attractor Network (RAN). RAN combines symbolic and subsymbolic features in a very attractive way: it is based on attractors; enables pattern classification under missing data; and demonstrates dynamic reconsolidation, which is very useful for tracking changing concepts. The perception RAN enables is somewhat reminiscent of human perception due to its context sensitivity. Furthermore, it enables an immediate and clear interface with symbolic memories, including loading of attractors by means of trivial wiring, updating attractors, and retrieving them faster without waiting for full convergence. It also scales to any number of concepts. This provides a useful counterpoint to more conventional memory systems, such as random access memory and auto-associative neural networks.

Keywords: Attractor dynamics; Memory; Reconsolidation; Context sensitivity; Perception

1. Introduction

Human memory is very different from current computer memory. Models of biological memories rely on dissipation and thus introduce attractors [35]. The use and application of dissipative dynamics in general analog computation was analyzed previously, with the attractors as either points, limit cycles, or chaotic [34,4,5]. Dissipative dynamics are considered to be the cause of persistent activity during memory experiments in both the Prefrontal Cortex and the Hippocampus [13,26,27,21,7,12,8,29]. Attractor neural networks were thus used or incorporated in many memory models [20,25,30,18,11,6,17]. These models typically assume fixed attractors to which the inputs flow, statically mapping the continuous input space into predefined basins of attraction. Experimental studies, however, suggest that neither the attractors nor the basins of their attraction in the input space are static; rather, they change upon retrieval via a process called Reconsolidation [9,31,22,10].

Perhaps the simplest psychological phenomena demonstrating the flexibility of attractors are the priming and gang effects. Priming is the phenomenon in which recently visited attractors have a higher chance to attract the next inputs [3]. In the gang effect, a gang of attractors has an effect beyond its own members; the visited attractors do not bias the landscape toward themselves only, but rather the pull of any attractor is affected by its neighbors' history as well [24].

The influence of the ordering of recently perceived inputs on the internal attractors in the Hippocampus was described by [29,23]. First [29] reported two distinct attractors in the firing of CA3 cells depending on whether a rat was put in a circle or square shaped environment. It also showed that when further morphed versions of the circle and the square environments were provided, the end states of the circle and the square became closer to each other. Interestingly, when the morphing was done in order from circle to square in small steps, the end representations became

* Corresponding address: Department of Computer Science, University of Massachusetts, Amherst, MA 01003, USA.
E-mail address: hava@cs.umass.edu.
URL: http://binds.cs.umass.edu/.

0167-2789/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.physd.2008.03.038

almost identical [23]. Similar phenomena was demonstrated in psychophysics experiments [32], where the order of the inputs was shown to influence recognition of faces. People were asked to identify previously presented faces from a test sequence of faces. Manipulation of the test sequence involved gradually transforming from one pre-learned face (source) to another initially distinguishable face (target). The results demonstrated partial merging of the source and the new face, but only when the faces were presented in gradual order from the pre-learned to the new face. A model based on the Hopfield network was proposed to explain this phenomenon [6]. Other existing neural networks also move their internal representations slightly in the direction of changing inputs, including networks used in competitive learning and in self-organizing maps [19]. All current neural networks models, however, suffer from a lack of practicality as a computational memory, since there are no existing procedures that can robustly translate a specification of attractors into a set of weights [36]. These memories are sub-symbolic and cannot be well coupled with symbolic items. Our goal is to create the foundation of a new kind of memory system that includes pattern completion in terms of attractor dynamics; that is easy to load, update, and retrieve; that is understandable and clear; and that still demonstrates the biological-like reconsolidation process along with its benefits for tracking changing concepts.

Recent work in neuroscience implies that attractors can be studied on different levels. Single cells were identified that recognize objects, people, and abstract concepts across fairly robust changes in modalities including the look, name, voice, etc. [33]. These were termed celebrity cells. Modeling attractors in this high level would mean allocating a node per attractor. A very practical model was suggested along this line called the "localist attractor networks" (LAN) in [36]. The LAN provides a unique combination of advantages: wiring the architecture to any given attractor landscape is simple; the network contains no spurious attractors, as in symbolic memories; and pattern completion and classification are available, as they are in neural network memories. Even the psychological features of gang and priming are demonstrated in the LAN.

In this paper we introduce the Reconsolidation Attractor Network (RAN), a generalization to the LAN that includes flexible memories, a controlled flow with early stopping, and contextual effects. The RAN's chief applicability is in computational paradigms which require pattern completion and tracking of changing concepts. Additionally, it can be applied to systems with any number of underlying prototypes. Due to its similarity to human memory, RAN will also be highly applicable in social robotics, where human-robot understanding is beneficial.

2. The reconsolidation attractor network (RAN) Model

RAN includes three types of nodes or cells: input cells (I), internal state cells (y), and attractor cells (A). Each attractor cell $A_i, i = 1, 2, \ldots n$ stores three pieces of information: the current activity or pull toward the attractor (a_i), the size of its basin of attraction (b_i), and the location of its center (c_i). The state cells y are the hidden variables that enable the convergence from the input to one of the given attractors, as in the LAN. The state cells change their values fast, reminiscent of neural network flow. In our model, the attractors are adaptable as well, but only when they are retrieved, as is the case in biological reconsolidation. We thus consider different time scales for the update of information within the system, which can be activity dependent. We note that multiple time-scales were shown to be history dependent at several levels of organization in the neural system, and they were considered to provide powerful means for computation and memory [16]. Our memory model demonstrates the applicability of this feature.

2.1. Contextual Effects in the State of RAN

The state cells update from the new input, when available, otherwise, they update recurrently based only on the state cells. The RAN creates contextual effects among the inputs by mixing previous states with input values to get the updated value of each state cell. The update equation of the state cells (vector y) is

$$y(t+1) = \mu \cdot y(t) + (1-\mu) \cdot \left[\alpha I + (1-\alpha) \sum_i a_i(t) c_i(t) \right] \quad (1)$$

where μ is the scalar describing the amount of context dependent memory, I is the input vector, and c_i is the center of the ith attractor. The vector c_i has the same length as the input and the state. The scalar α is the pull of the state toward the input versus the attractor. The value of α is typically decreased between the introduction of new inputs. Other protocols that attend to the inputs will be described in Section 3.3. The (scalar) activity of the attractor a_i is a function of its distance from the state, normalized by the size of its basin:

$$a_i(t) = \frac{d[y(t), c_i(t), b_i(t)]}{\sum_j d[y(t), c_j(t), b_j(t)]}. \quad (2)$$

It was assumed [36] that all attractor basins are equal and update by $b^2 = 1/n \sum_j a_j(t) |y(t) - c_j(t)|^2$; we follow a similar assumption. The function $d[\cdot]$ measures the distance between the state and the attractors. If we chose $d = e^{[-(y-c_i)^2/2b^2]}$, then the attractor cells would be similar to a layer of radial basis functions with normalization [28].

2.2. Entropy of attractors' activity distribution

When is it appropriate to stop the iteration of the state nodes in order to make decisions? In theory, we could wait until the state nodes reach an attractor. However, stopping early prevents over-fitting and correlates better with biological systems which require a timely response. We propose to employ an entropy threshold based on the distribution of attractors' activities:

$$H_a(t) = \sum_{i=1}^{n} a_i(t) \cdot \log_2 \left(\frac{1}{a_i(t)} \right). \quad (3)$$

According to our definition of activity entropy, the algorithm is guaranteed to have non-increasing entropy with subsequent iterations. If not stopped the state will always converge to the attractor with the lowest possible entropy value. However, with early stopping the state nodes will end the updating process when entropy is small enough, which occurs at the peak of the distribution of the attractors' activity. At that point the algorithm will be able to identify the attractors closest to the current state, and the attractors will update based on that state. As the activity of an attractor is a function of its center and its basin, individual changes to b_i could bias the stopping landscape. Individual changes of basins will be considered in a future study.

In addition to the use of a low level of the activity entropy for early stopping of the state iteration process, we will also consider a high level of the entropy. Very high entropy means that no existing attractor can explain the input. We propose in Section 4.1 a general framework of memory where high entropy indicates that the current memory models held in short term memory are insufficient to explain the input. This creates a need to retrieve or create new memories. This use of entropy is related to the mathematical modeling of the surprise raised by a new input relative to the current internal landscape [2]. Our notion better evaluates the peaked distributions, and it thus provides a decision value for early stopping.

2.3. Updating the attractors: Reconsolidation

It is proven [36] that the LAN's dynamic corresponds to a search for a maximum likelihood interpretation of the observation. This holds true for our networks even though we add the tuning of the interpretation. RAN's attractors can be updated when the activity distribution peaks, as measured by the entropy. If we consider the attractors as generative models then the center should represent the mean of the observed inputs. This is possible if learning is performed using an unbiased estimator. We thus employ the following update:

$$c_i(t+1) = \nu a_i(t) \cdot y(t) + [1 - \nu a_i(t)] \cdot c_i(t) \tag{4}$$

where ν is the flexibility coefficient controlling the amount of lability in the attractors. The extreme value of $\nu = 0$ results in no reconsolidation, and the value of $\nu = \frac{1}{a_i}$ causes the attractor to fully change to the final internal state. Note that in our system, all active attractors can update, while the most active ones update the strongest.This joint updating will cause attractors that are frequently active together to become even closer, while ones that are active at different times will move further apart. Such processes of merging and separation are achievable simply by setting the stopping entropy value to the levels of zooming in or out of details, as desired.

3. Properties of the RAN: Simulations

The chief advantage of the RAN over the LAN is its reconsolidation property. There are no changes to already existing gang and priming effects, and the RAN is as simple and scalable as the LAN and with no spurious attractors. In this section we provide examples of the type of reconsolidation available in the RAN. While the simulations are minimalist, they adequately express the network's behavior.

3.1. Attractor reconsolidation: Growing a beard

The data described in [23,32] suggest that an attractor is updated following an input sequence consisting of monotonic changes from an input that fits an attractor to a foreign input. We demonstrate this phenomenon in the RAN and describe our findings.

For this simulation we wire the system with three attractors which represent different faces: circular (Frank), semi-circular (Nate), and square (Stu). The inputs are 2D gray scale matrices with 237×237 values, and the internal states of the network are similarly represented in the pixel domain. Naturally, there is much overlap in the state cells of the different faces but the attractor cells do not overlap, according to our construction (as is the case in celebrity cells).

Next, the RAN is presented with an input sequence in which the circle face grows a beard in seven small steps. Fig. 1b depicts the distance of each attractor node from each input, when the attractors are forced to remain fixed (as in LAN) with no reconsolidation ($\nu = 0$). As Frank's beard grows, we see that the distance between the input and each attractor increases, and that the biggest relative increase in distance occurs in the attractor of Frank's original face. We next relax the flexibility control and let the attractors reconsolidate. In Fig. 1c we see the distance of the attractors from the same sequence when the attractors can update with the inputs received. We include eight learning steps for each input in which the attractor is pulled towards the input. The RAN modeling allows the hypothesizing of what the new attractors represent as shown in Fig. 1d, which has not been explicitly suggested before. We see that as the circle face grows a beard monotonically, the attractor of the circle face changes to become a bearded face, the semi-circular attractor adjusts modestly, and the square face which is farther away does not update at all.

In Fig. 2 we run the same experiment but with different values of the entropy in the stopping condition. Higher entropy in the stopping criterion causes bigger changes to near-by attractors. This is because the distribution of the attractor activity was not highly peaked, and the activity of closer attractors is not significantly different from the activity of the winning one. A lower entropy condition halts the update of the internal nodes in a more peaked distribution, and, thus, an attractor that does not win has a much lower activity and is affected very slightly by the input. When shuffling the input set of bearded face images such that they are not provided in the previously used monotonic order, the attractors only adjusted mildly, demonstrating the importance of the monotonicity of input as seen in [23].

3.2. Contextual Perception: SOS versus 505

We next focus on how RAN causes biases in perception; this is indeed a chief property of human memory. We demonstrate the bias by a system that will read SOS even when it is not

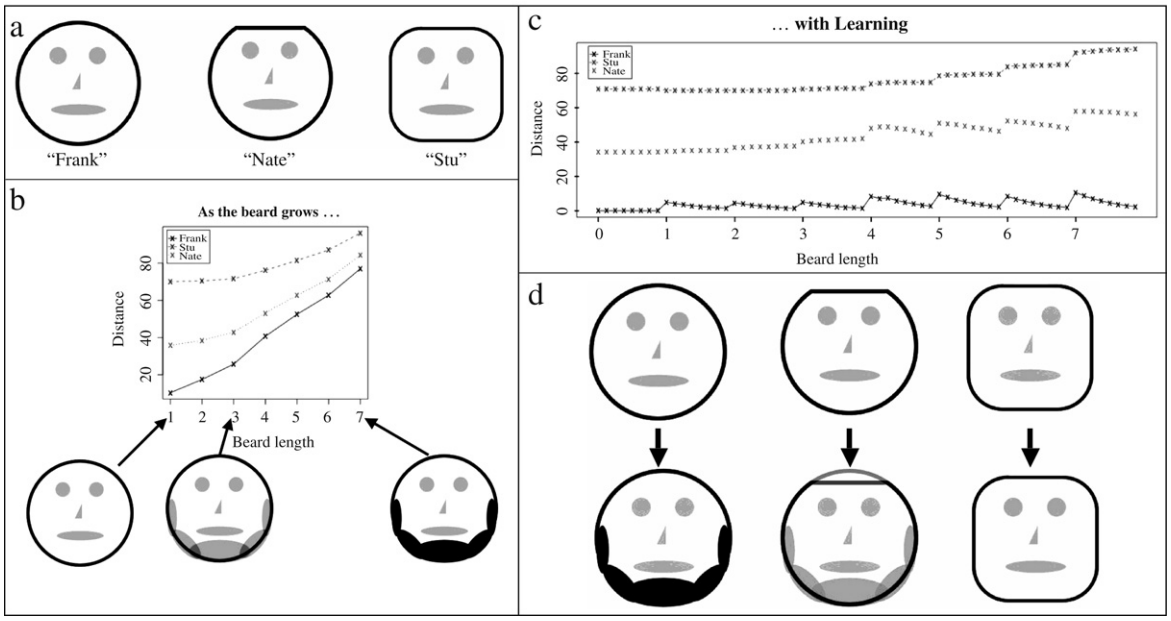


Fig. 1. Memory concepts change with a monotonic input sequence that leads toward a new concept: (a) Three faces are stored as non overlapping attractor memories (b) Seven inputs arrive sequentially featuring Frank growing a beard. The distance of each attractor from each input is depicted for when the attractors are held static. The Frank attractor increases its relative distance from bearded Frank (c) The distances of the three attractors from the seven inputs when attractors are flexible (d) The modified attractors are depicted: Frank changes to a bearded Frank, Nate will recognize both clean shaved and bearded Nates, and the Stu memory has not been modified.

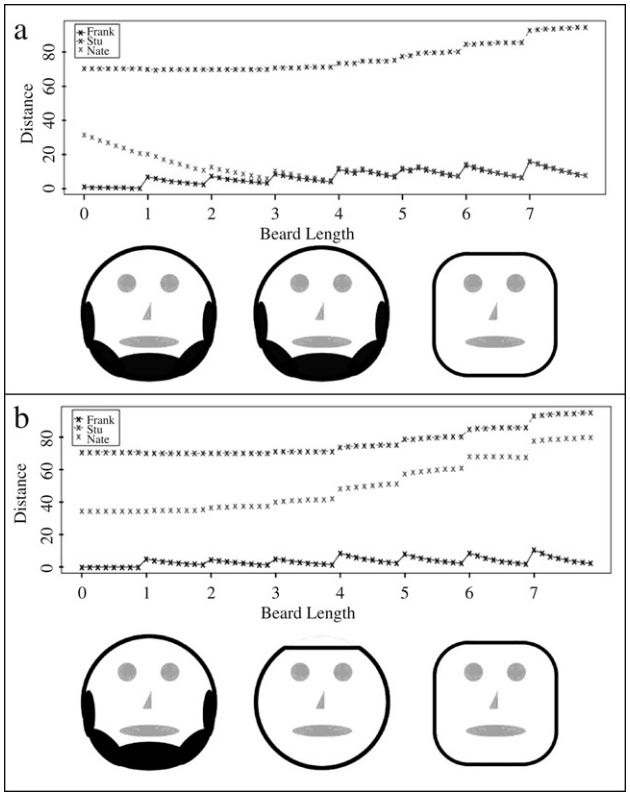


Fig. 2. Updating related attractors. (a) When entropy is set high for early stopping, the attractor activity distribution is less peaked, and neighbors of the most peaked attractor will update their values as well. Here the stopping condition is set to $H = 1$. (b) Memory updates better focus on the winning attractor when a lower level of the entropy (higher peak) is required as the stopping condition. Here the stopping condition is set to 0.25.

exactly written there, based on context and expectation. We focus on the effect that occurs due to persistent continuous activity. To do this, we hold the attractor landscape unchanged during the experiment.

We initiate a RAN with four attractors: the letters S and O, and the digits 0 and 5. The inputs to the network are based on 20 point bold face font represented by a 25×25 pixel image and an additional column which specifies whether the image is a letter (the whole column is 1) or a digit (the whole column is 0). In addition to true letters and digits, we also formed an input image containing a combined morphing of 5 with S (50% each) concatenated with a column vector with values of $\frac{1}{2}$. Analogously, we made another image by morphing 0 with O (50% each) and appending an identification column of all $\frac{1}{2}$. These inputs were used for testing. While in the previous demonstration we visualized the attractors themselves, we now want to visualize the flow of the state nodes within the attractor space. To visualize the attractor space (where the attractors are only points) we use Principle Component Analysis (PCA) across the state space on the full images, including the image pixels and the identification columns. Fig. 3a shows the attractors mapped to a 2D space after applying the PCA. The digit 5 lies in the top right area, the letter S is in the bottom right; both the digit 0 and the letter O are on the left side of the space but the digit lies higher than the letter. Because of the identification column, the space is divided into the letters area and the digits area, although we may not be able to completely see this division.

The first demonstration includes the sequence of three consecutive inputs: the letter 5, the morphed image of O-0, and the morphed image of S-5. The system first recognizes the 5; then, with the next input, it flows to the digit 0; and

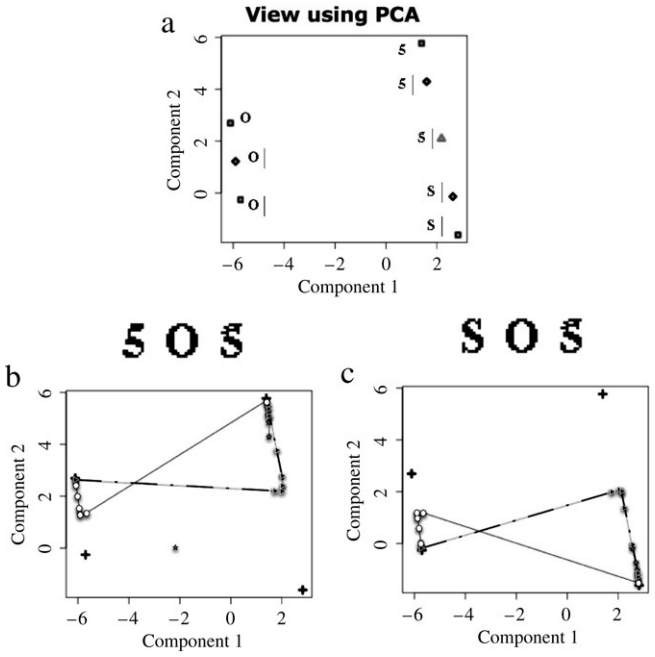


Fig. 3. Contextual effect due to persistent continuous activity in the state nodes is demonstrated by reading SOS or 505 from similar sequences based on the starting attractor. The attractors correspond with the digits 0 and 5 and the letters O and S. Also considered as input are combinations of 0-O and 5-S. The representation of each digit and letter includes both the pixel image as well as an identification column of 1's for a letter, 0's for a digit, and a fraction when it is unknown. (a) The high dimensional space of the letters and digits is viewed in 2D via PCA applied to the image concatenated with the identification column. (b) The input sequence is the digit 5 followed by 50% of 0-O and then by 50% of 5-S. The flow after the presentation of the first digit is depicted with gray stars, the flow after the presentation of the 0-O are white circles, and the flow after the presentation of the third input are black triangles. It can be seen that the trajectory flows to an unstable middle point first, and then it is biased toward digits (the first attractor). (c) The input sequence is the letter S followed by 50% of 0-O and then by 50% of 5-S. The flow is depicted with stars, circles, and triangles after the representation of the first, second, and third inputs accordingly. As in part (b), the trajectory also flows to an unstable middle point first, but then it is biased toward letters (the first attractor in this case). The bias occurs since the state nodes leave traces of their previously seen inputs, which act as the prior bias to perception for the next input signals.

with the last input, it converges to 5 (see Fig. 3b). The fact that the first input was a digit biased the following perception; this is a generalization of the priming effect. In the control demonstration, the system was first shown the letter S and then the same two morphed images of O-0 and S-5, as in the previous manipulation. The system now reads SOS. This occurs because the internal state is not wiped after the recognition of an attractor ($\mu > 0$), and it leads to the continuous activity of the state space that causes contextual effects. Note that we kept the attractor's state unchanged in order to show that continuous activity can bias a percept. In reality the effect of such a bias on perception is even stronger since, as 5 and 0 occur more frequently together, the joint reconsolidation process strengthens their link to each other.

3.3. Input dynamics and continuously many attractors

In all previous experiments the input affected the state nodes in exponentially decreasing amounts (α) after the first presentation. It is possible however to consider the input in different attention levels during the process of state flow. We will study different protocols of lingering on input, and this will demonstrate that different attention protocols will change both the dynamic and the end state of the network. In other words, both the flows and the attractors that are reached will differ.

For this study we introduce the RAN with three attractors, each of which represents a line with a given length and rotation degree from the vertical direction (see Fig. 3c): the first attractor is a line that is presented with a 0 degree rotation and a 45 pixel length (0, 45), the second one is a line with a 90 degree rotation and a 45 pixel length (90, 45), and the third one has a 45 degree rotation and is only 10 pixels long (45, 10). The inputs and states correspond to lines as well. The input itself is a set of pixel images, but for simplicity of presentation we show them in the 2D space of rotation and length where the attractors are points.

We experimented with four update protocols for α. Since, α affects how much the input is considered, this can be perceived as studying attention protocols. In the constant attention protocol, the input gets a constant weight. For linear and exponential, the input is considered in decreasing amounts for a few steps and then jumps up again; and in the periodic protocol, the input is considered and then not considered at all and then is considered again, etc., see Fig. 4a. We see in Fig. 4b that for input (44, 20) the constant and exponential protocols lead to fixed points, while the linear and exponential ones lead to limit cycle dynamics in state space.

In Fig. 4d we attend to the input constantly but to varying degrees, which bring about different fixed point attractors in the state nodes. Note that while all the attractors at the state space of this demonstration are close to the true attractor (0, 45), none fuses with it. In Fig. 4e we consider two different strengths of inputs for the periodic protocol, and we see that a small change in the strength of the periodic signal can cause a totally different flow in the state space. In one case, the state flows to a fixed point and in the other one it flows to a limit cycle attractor.

When a series of different inputs is considered the slight changes in the protocols may lead to different associations of RAN's closest attractor. The input of (30, 25) when presented after (44, 20) reaches a different attractor than when it follows (46, 20). In the first case it goes to attractor (0, 45), and in the second case it goes to attractor (45, 10), as seen in Fig. 4f.

This demonstration shows that a network with a discrete number of fixed point attractors may lead to a continuum of fixed point and limit-cycle attractors in the state node space based on differences in attending to the input. Also, the path to convergence of the state nodes can be very complex and varies for the different protocols. The perception and consequently the memory of the input can differ based on the protocol considering the inputs.

4. Discussion

We present a computational technique based on Reconsolidation Attractor Networks (RANs) for modeling various aspects of memory storage, retrieval, and reconsolidation. The

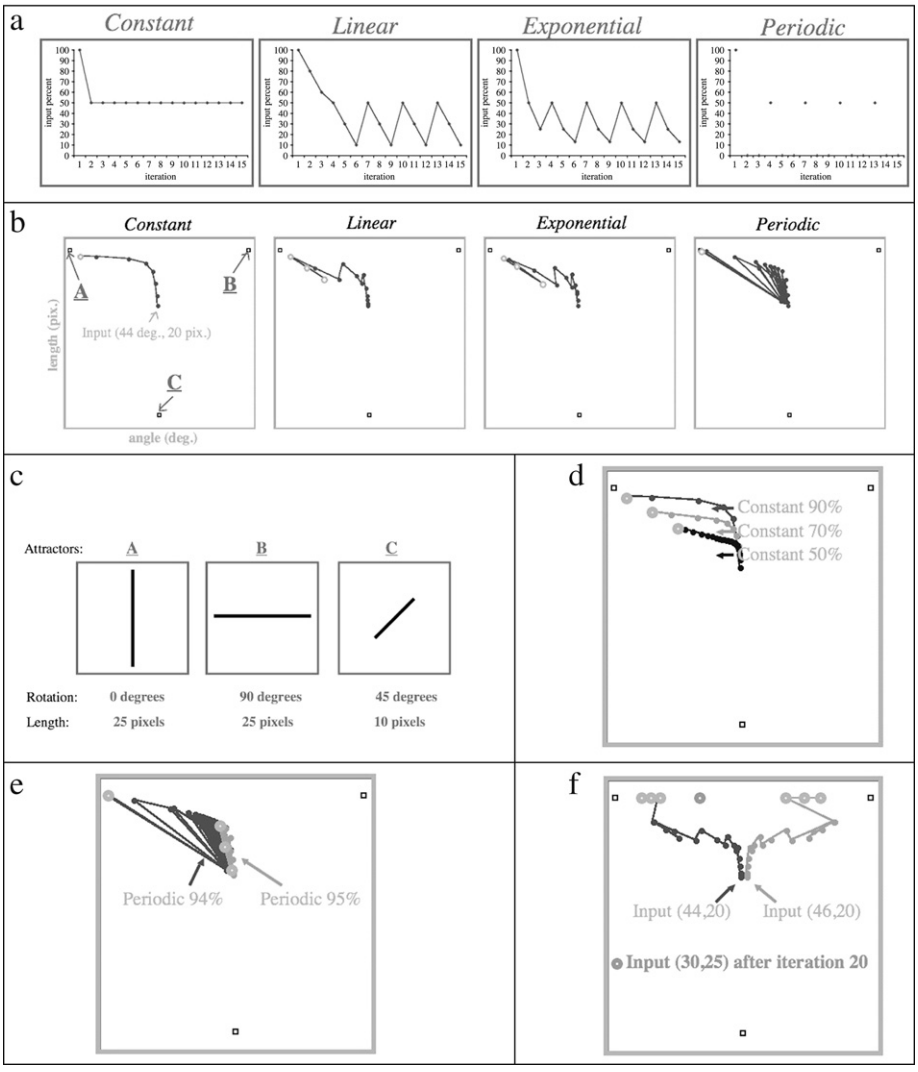


Fig. 4. We study the flow in state nodes and how it varies with different attendance to the input: (a) Four different protocols of controlling α in Eq. (1) are considered (b) The different protocols lead to different trajectories on input of (44 degree rotation, 20 pixel length) (c) The three attractors of the RAN are the three lines. Here we show the attractors in pixel form as appeared in the input and node spaces; in all other figures the inputs are shown in the 2D analogous representation of degree-length state space in order to view the attractors as points (d) Three different point attractors in the state node space are reached when the input is considered in the constant protocol but with varying degrees; none merge with the original attractors (e) Different trajectories and type of attractors result in considering the input in the periodic protocol and with 1% difference in strength (f) This figure demonstrates contextual effects. First the input (44, 20) is introduced, followed by (30, 25). The trajectory starts with a limit cycle after the introduction of the first input, and it flows to a fixed point close to the attractor (0, 45) after seeing the second input. In the other case, an input of (46, 20) leads to a different limit cycle, and with the introduction of (30, 25) the flow goes to a fixed point near (45, 10).

chief focus is on unconventional memory for use in computational systems. In the RAN, raw input is translated into dynamic activity across a state space, whose activity in turn is influenced by the activity levels across various attractors. The attractors themselves can be modified during reconsolidation, which is triggered by an entropy criterion that kicks in when the distribution of activity across the various attractors becomes sufficiently sharp. The model accounts in an abstract way for certain contextual memory effects by showing that RANs exhibit hysteresis. In other words, the system settles into a different attractor for a given input depending on the sequence of prior inputs. Such hysteresis can result either from persistent activity across the state space or from the movement of the attractors themselves. In our model, attractor centers can be morphed semi-continuously by gradual changes in inputs. The RANs continue to function sensibly even at an arbitrarily large number of attractors, since the attractors can be defined in a higher dimensional space and keep distance by this operation. For example, we can think of a 200 dimensional input with 200 attractors that lie on the corners of a 200 dimension hypercube, so that they are spaced out [36]. It is possible that during reconsolidation the attractors will become closer or further apart based on the level of entropy chosen for early stopping of the peaked distribution.

The practicality of the RAN over neural network attractor modeling lies in its ability to bridge symbolic and subsymbolic information naturally; it is possible to load any set of symbolic data by simple wiring, a feature which is not available at all

in neural networks. A chief superior feature of our memory is its ability to reconsolidate, which means it can track changing concepts. The use of entropy and the update of prototypes similar to the winning one are supportive of generalization and transfer learning.

It is our intention to continue to study the practicality of reconsolidation based memories, in particular the tracking of dynamic concepts while maintaining robustness and generalization. We will also explicitly model the merging of prototypes and their hierarchical decomposition. We will focus on individual changes to the basins of the attractors and study how this will modify prototype selection.

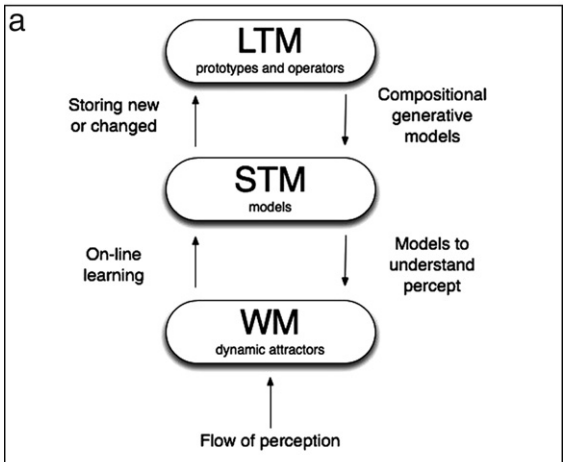
4.1. The large view of memory system

While our model may be removed from neurological structure, it still has some functional similarity. Anatomically, it has long been recognized that the initial site of learning is different from the eventual site of storage. Hippocampal lesion experiments demonstrate that the hippocampus is necessary to learn a new memory, as well as to recall it in a relatively short period of time after learning. After some time, hippocampal lesions no longer disrupt previously acquired memory. However this is not the case when memories are recalled, as they then rely on the hippocampus for the process of reconsolidation.

This leads us to propose the RAN as part of the memory system. According to this view, long term memory may include prototypes (e.g., ball, red, dark, line, degree, loud) and operators that can be applied to prototypes, where operators may also include associations between different prototypes and return concrete concepts or models (e.g., a red ball). Operators also enable longer and more complicated sequences of other prototypes. The resulting models are sent to short term memory, and they are used as the reference of the working memory, which can receive input and perceive it according to the models in short term memory. The state nodes of the RAN correspond to working memory, and the individualized attractors correspond to the models in short term memory. According to the RAN, the models of STM can be modified in accordance with the new inputs. The change to LTM occurs on a much slower time scale and directly influences the individual prototypes and operators; it is not part of the current RAN model. When input arrives, if the entropy of the STM attractors with respect to the input is too large, no memory model is able to explain the input, and new memory models are requested to be composed from LTM into STM. Fig. 5 explains our computational and algorithmic view of memory.

4.2. RAN in Future Modeling and Predictions

Unlike neural networks, RAN's parameters are easier to link to their corresponding psychological values since they relate to the center and pull of attractors. This leads us to suggest using RAN both to predict results and to design biological experiments based on the computer experiments done in this work. Based on the first experiment of growing a beard, we hypothesize that reconsolidation affects the priors of not



b

Algorithm 1 The Memory Reconsolidation framework

Require: a new percept arrives
  Consider the models in STM
  if STM is not relevant {as measured by too high entropy}) **then**
    Generate new models to STM from LTM
  else {entropy is not too high}
    while percept is well explained or time limit passes **do**
      Update WM using percept and models
      Update model likelihoods using WM state
    end while
    Improve models in STM based on the WM
  end if
  Only once in long time Improve some prototypes or operators in LTM

Fig. 5. Algorithmic view of memory system: (a) The figure depicts Long Term Memory that includes prototypes (e.g., ball, red) and operators (e.g., association) that together can form instances or models that are deposited into Short Term Memory. Short Term Memoryś models can be updated by Working Memory via the RAN update algorithm. Working Memory hosts the state nodes of the RAN and enables the perception of new inputs. (b) The algorithm associated with the proposed framework of memory.

only the particular memory but also of close memories. This prediction can be tested in psychophysics. Humans subjects will be asked to categorize sets of stimuli. The stimuli will be generated from a set of prototypical random stimuli with added noise, so the prototypes are never presented without noise. First, subjects will be trained to categorize the patterns, and then changes will be applied to one underlying prototype. The learning rate will be measured. Later, similar changes will be made to another category, and the learning rate will be compared with the first one. The rate of learning will be estimated by both the percentage of correct classification and the reaction time [1].

The second experiment will check the hypothesis that subjects bias their view to recent associations. First, a sequence of images will be shown that either bias toward only digits or only letters, then distorted images will be shown, each combining an image of a digit with an image of a letter. The subjects will be asked to type what they see. As a second manipulation, successions of the letter S and the digit 5 will appear frequently together, and then the blurred test images will be shown again.

The third experiment will measure the effect of attention via both color saliency and explicit directions. These will be applied to the first experiment described above. During the experiments the RAN will be compared with data. One may

want to also compare the model with evidence of multi-scale and sharp learning [14,15].

To summarize, the main focus of this contribution is the design of memory that can be used for improved thinking machines that are able to follow dynamically changing concepts and demonstrate sensitivity to context. The new computational machines will naturally combine learning from examples, high-level directions, and cognitive attention and, thus, will change the state of the art of machine learning, which is currently best equipped to produce rigidly single task oriented algorithms.

Acknowledgments

This work is supported by the ONR grant #$N00014-07-1-0009$. The author thank Lars Holtzman, Ueli Rutishauser, Yariv Levy, Megan Olsen, and David Cooper for assistance and advice.

References

[1] F.G. Ashby, W.T. Maddox, Human category learning, Annu. Rev. Psychol. 56 (2005) 149–178.
[2] L. Itti, P. Baldi, Bayesian surprise attracts human attention, Adv. Neural Inform. Process. Syst. 19 (2006) 547–554.
[3] S. Becker, M. Moscovitch, M. Behrmann, S. Joordens, Long-term semantic priming: A computational account and empirical evidence, J. Exp. Psychol.: Learning Memory Cognition 23 (5) (1997) 1059–1082.
[4] A. Ben-Hur, J. Feinberg, S. Fishman, H.T. Siegelmann, Probabilistic analysis of a differential equation for linear programming, J. Complexity 19 (4) (2003) 474–510.
[5] A. Ben-Hur, J. Feinberg, S. Fishman, H.T. Siegelmann, Random matrix theory for the analysis of the performance of an analog computer: A scaling theory, Phys. Lett. A. 323 (3-4) (2004) 204–209.
[6] B. Blumenfeld, S. Preminger, D. Sagi, M. Tsodyks, Dynamics of memory representations in networks with novelty-facilitated synaptic plasticity, Neuron 52 (2006) 383–394.
[7] M.V. Chafee, P.S. Goldman-Rakic, Matching patterns of activity in primate prefrontal area 8a parietal area 7ip neurons during a spatial working memory task, J. Neurophysiol. 79 (1998) 2919–2940.
[8] A. Compte, N. Brunel, P.S. Goldman-Rakic, X.J. Wang, Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model, Cerebral Cortex 10 (2000) 910–923.
[9] Y. Dudai, Time to remember, Neuron 18 (1997) 179–182.
[10] Y. Dudai, M. Eisenberg, Rite of passage of the engram: Reconsolidation and the lingering consolidation hypothesis, Neuron 44 (2004) 93–100.
[11] D. Durstewitz, J.K. Seamans, T.J. Sejnowski, Dopamine mediated stabilization of delay-period activity in a network model of prefrontal cortex, J. Neurophysiol. 83 (3) (2000) 1733–1750.
[12] C.A. Ericson, R. Desimone, Responses of macaque perirhinal neurons during and after visual stimulus association learning, J. Neurosci. 19 (1999) 10404–10416.
[13] J.M. Fuster, G. Alexander, Neuron activity related to short-term memory, Neuron 14 (1971) 477–485.
[14] C.R. Gallistel, S. Fairhurst, P. Balsam, The learning curve: Implications of a quantitative analysis, PNAS 101 (30) (2004) 13124–13131.
[15] C.R. Gallistel, T.A. Marka, A.P. King, P.E. Latham, The rat approximates an ideal detector of changes in rates of reward: Implications for the law of effect, J. Exp. Psychol.: Animal Behavior Processes 27 (4) (2001) 354–372.
[16] G. Gilboa, R Chen, N. Brenner, History-dependent multiple-time-scale dynamics in a single-neuron model, J. Neurosci. 25 (2005) 6479–6489.
[17] A.J. Gruber, P. Dayan, B.S. Gutkin, S.A. Solla, Dopamine modulation in the basal ganglia locks the gate to working memory, J. Comput. Neurosci. 20 (2006) 153–166.
[18] M.E. Hasselmo, E. Schnell, E. Barkai, Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region ca3, J. Neurosci. 15 (1995) 5249–5262.
[19] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1999.
[20] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. USA 79 (1982) 2554–2558.
[21] L.M. Kay, L.R. Lancaster, W.J. Freeman, Reafference and attractors in the olfactory system during odor recognition, Int. J. Neural Systems 7 (4) (1996) 489–495.
[22] J.L.C. Lee, B.J. Everitt, K.L. Thomas, Independent cellular processes for hippocampal memory consolidation and reconsolidation, Science 304 (2004) 839–843.
[23] J.K. Leutgeb, S. Leutgeb, A. Treves, R. Meyer, C.A. Barnes, B.L. McNaughton, M.B. Moser, E.I. Moser, Progressive transformation of hippocampal neuronal representations in morphed environments, Neuron 48 (2005) 345–358.
[24] J.L. McClelland, D.E. Rumelhart, An interactive activation model of context effects in letter perception: Part i. An account of basic findings, Psychological Rev. 88 (1981) 375–407.
[25] B.L. McNaughton, R.G.M. Morris, Hippocampal synaptic enhancement and information storage within a distributed memory system, Trends Neurosci. 10 (1987) 408–415.
[26] Y. Miyashita, Neuronal correlate of visual associative longterm memory in the primate temporal cortex, Nature 335 (1988) 817–820.
[27] Y. Miyashita, H.S. Chang, Neuronal correlate of pictorial short-term memory in the primate temporal cortex, Nature 331 (1988) 68–70.
[28] T. Poggio, F. Girosi, Regularization algorithms for learning that are equivalent to multilayer networks, Science 247 (1990) 978–982.
[29] T.J. Wills, C. Lever, F. Cacucci, N. Burgess, J. O'Keefe, Attractor dynamics in the hippocampal representation of the local environment, Science 308 (2005) 873–876.
[30] A. Treves, E.T. Rolls, Computational constraints suggest the need for two distinct input systems to the hippocampal ca3 network, Hippocampus 2 (1992) 189–199.
[31] K. Nader, G.E. Schafe, J.E. Le Doux, Fear memories require protein synthesis in the amygdala for reconsolidation after retrieval, Nature 406 (2000) 722–726.
[32] S. Preminger, D. Sagi, M. Tsodyks, Morphing visual memories through gradual associations, Perception Suppl. 34 (2005) 14.
[33] R. Quian Quiroga, L. Reddy, G. Kreiman, C. Koch, I. Fried, Invariant visual representation by single neurons in the human brain, Nature 435 (2005) 1102–1107.
[34] H.T. Siegelmann, S. Fishman, Computation by dynamical systems, Physica D 120 (1998) 214–235.
[35] H.T. Siegelmann, A. Ben-Hur, S. Fishman, Computational complexity for continuous time dynamics, Phys. Rev. 83 (7) (1999) 1463–1466.
[36] R.S. Zemel, M.C. Mozer, Localist attractor networks, Neural Comput. 13 (2001) 1045–1064.