Spinning brain

An interactive program for associative recall of visual patterns

R. Kühn[1] and G. Sawitzki[2]

Abstract

Recent work of Hopfield has revealed how many powerful techniques of condensed matter physics can be applied to problems arising in brain theory. One of the novel aspects is that brain functions such as recalling a memory (a visual pattern or a tune) can be modeled as *collective* phenomena. In these developments, the role of numerical simulations has been substantial both in confirming the analytical work and in testing practice-oriented algorithms. Here we discuss the architecture and performance of an interactive program, which has been devised to simulate storage and associative retrieval of visual patterns. The paper also provides ample background information so as to allow the reader to interpret the program in the proper context of these fascinating recent developments.

1. Introduction

No doubt, in many respects the brain is superior to a computer. Elementary tasks, such as the completion of fragmented visual patterns and the recollection of structured data, are performed by the brain with great ease and efficiency. This is partly due to a high degree of parallel operation, partly to a different organization of the "hardware". The brain can be considered a surprisingly efficient associative, or content-addressable, memory. If, for instance, you offer the brain a pattern which is (not too) distorted, then it is able to recognize and reconstruct the original *without* checking all the other ones it may have in memory. Thus, in contrast to what is done in present day computers, the information is not classified by label but by its content - whence the name.

In this paper we describe an interactive program for the associative recall of visual patterns. In so doing we take the opportunity to sketch some underlying ideas from the theory of neural nets. The paper is organized as follows. In Section 2 we review some current ideas about biological memories and their formalization. Then we discuss a model due to Hopfield (1982), which has received a huge amount of interest, in particular among physicists. In Section 4, the program itself and its performance are analyzed. Finally, we indicate some extensions in Section 5. The conclusion (Section 6) is that there has been substantial progress in brain theory, certainly in the understanding of *collective* aspects, but that nevertheless the still remaining distance between modeling and reality should leave us rather modest.

Though we have, in what follows, refrained from including an extensive list of references, it goes without saying that the neural network theory as reviewed in the present paper is a joint achievement of various groups from all over the world. To give proper credits and to obtain further information, the interested reader may wish to consult recent review articles by Domany (1988) or those collected in the proceedings of the *Heidelberg Colloquium on Glassy Dynamics* (van Hemmen and Morgenstern, 1986).

[1] Sonderforschungsbereich 123, Im Neuenheimer Feld 294, D 6900 Heidelberg

[2] 2 StatLab, Im Neuenheimer Feld 294, D 6900 Heidelberg

Part of the work on neural nets was done in the group of J.L. van Hemmen and H. Horner at the Universität Heidelberg (SFB 123). It is a pleasure to thank the participants, in particular B. Sulzer, for their help and advice. The program described in Section 4 is based on development tools of the StatLab Heidelberg.

2. What do we want to model?

Models of biological memories which are based on customary storage procedures for computers suffer from two major constraints. First, storage and retrieval algorithms usually employed for computer memories are inefficient and (orders of magnitude) too slow when applied to important tasks such as pattern recognition, which are of vital importance in daily life. Second, assuming individual neurons to be responsible for storing a single individual item results in the need for an unrealistically large number of neurons as soon as realistic problems are to be handled. Besides this, no conventional storing mechanism has been found up to now that enjoys enough flexibility and fault tolerance to serve as a plausible model for biological memories.

Further evidence disproving a strictly localized representation of knowledge in the brain arises from neurophysiological experiments. Injuries of the cortical area, which is responsible for information storage, do not lead to a total loss of the stored items, but only increase the number of retrieval errors and inaccuracies. Moreover, if the brain is offered sensorial stimuli and it has to decide whether or not the resulting perception corresponds to a stored item, it need not perform an extensive search through all the memories. Rather is the *retrieval process controlled by attributes of the sensory input.* Therefore, a biologically relevant type of memory has to be *content-addressable* (cf § 1) and provided with a holistic, distributed representation of knowledge.

If, then, information is not stored in the neurons, what else can the system use to deposit the data? The widely accepted answer, which is surprisingly simple, was given by D. O. Hebb in his book "The organization of behavior" (1949). Information is transmitted between neurons via the synapses and according to Hebb, precisely the synapses carry the data presented to the system during the learning sessions. In simple terms, the Hebb rule states: When a neuron significantly often takes part in firing another neuron, this regularity is "detected" by the synapse connecting the two neurons, and as a consequence its efficacy increases. A very attractive feature of this learning rule is the *locality* of the storage mechanism. Any synapse only has to recognize the events in its immediate neighbourhood, the two neurons it connects. No information about the global state of the network is required for the synapse to do its job properly.

The Hopfield model, to which we turn shortly, employs the following two ingredients. The data are stored in the synapses according to the Hebb rule and the neurons are simplified, i. e., they are assumed to have only two states, quiescent and active. This kind of so-called *formal* neuron was first hypothesized by McCulloch and Pitts (1943). They showed any computable task can be performed by a suitably designed system of formal neurons.

3. The Hopfield model

a. Structure and dynamics

In the Hopfield model each neuron S_i is supposed to acquire only two states: either firing at maximum rate or being quiescent. The actual state of a network of N neurons can then be described as an N-dimensional vector whose components S_i take the values +1 or -1 (firing or non-firing). Because of their two-state nature, the neurons are formally equivalent to certain models of magnetic moments, known to physicists as *Ising spins.* The patterns to be stored are certain Ising spin configurations. In Hopfield's modeling (1982) the learnt patterns are actually chosen in such a way that the components are generated by a random-number generator, taking the values ± 1 with equal probability. This assumption is important for analytical work but can be dropped in practice - as shown in Sec. 4, where we present a description of our program "Spinning brain". To fully appreciate the intentions (and restrictions) of this novel type of argument, we will first analyze the Hopfield model in some detail.

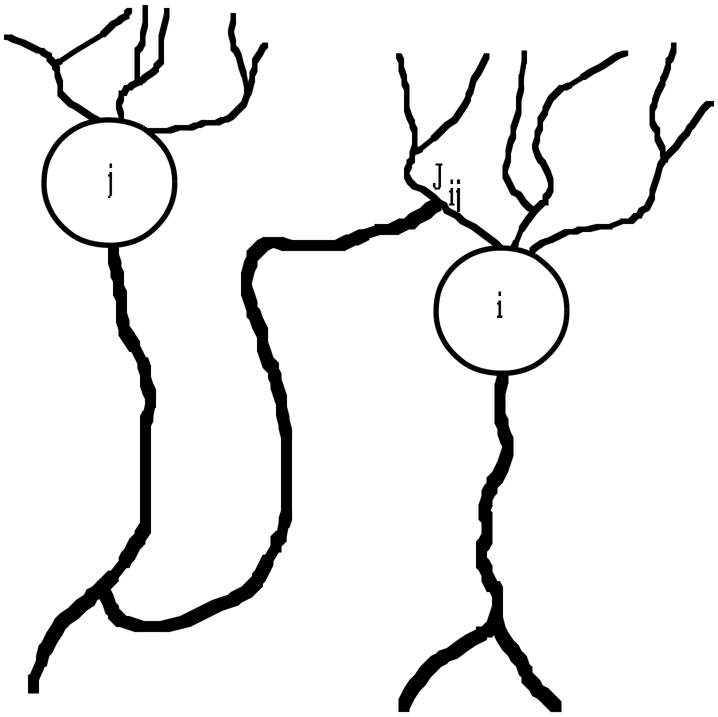


Fig.1: Schematic view of two interacting neurons. Circles represent cell bodies, thin lines dendritic trees, and bold lines the axons of neurons *i* and *j,* respectively. A synapse of efficacy J_{ij} is shown where *j'*s axon connects to *i'*s dendritic tree. The contribution of neuron *j* to the postsynaptic potential of neuron *i* is $J_{ij}S_j$.

The interaction of the neurons in the network is accomplished by the synapses whose efficacy in our context is described by the couplings J_{ij}. In the initial state before learning, all couplings (bonds, synaptic efficacies) are set to zero. If a new pattern, say μ, is learned, the bonds change in the following way. 1/N is added to J_{ij}, the synaptic efficacy for transmissions of signals from neuron j to neuron i, if in pattern μ neuron i is in the same state as neuron j; otherwise 1/N is subtracted. This instruction yields *symmetric* bonds. It is a particular formalization of the Hebb rule. The bonds are no longer modified when the learning session is over. Thus, in some way the network is to be told whether it is in a learning or in a retrieval mode.

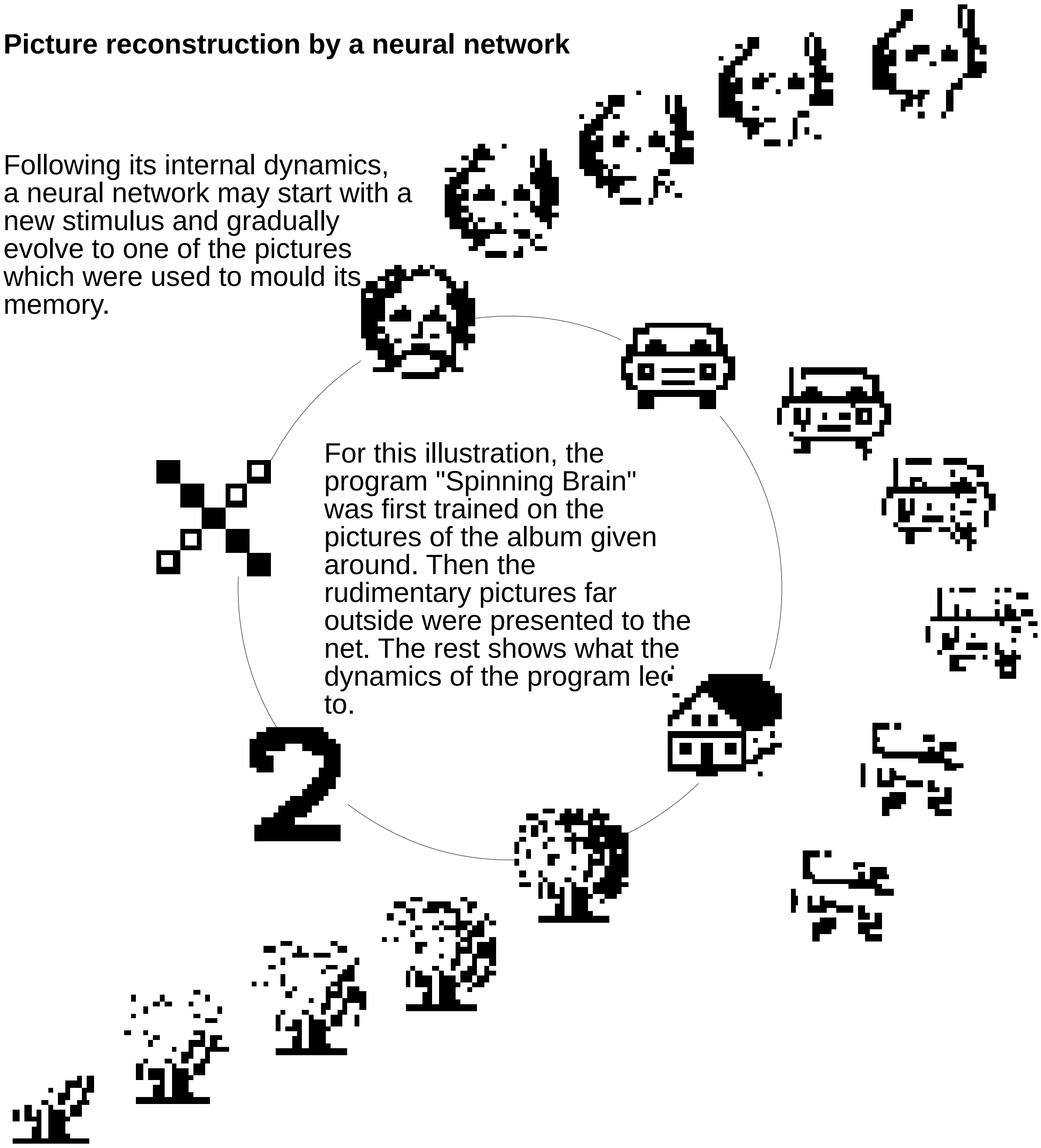
Next we focus on the operational characteristics of the network during retrieval when a stimulus with components S_j is presented. At any time, each neuron collects the information it receives from all the other neurons. This is the postsynaptic potential or local field, $h_i=\sum_j J_{ij}S_j$. The neuron determines its subsequent state according to the value of its postsynaptic potential h_i. A possible prescription is that neuron i will fire, if h_i exceeds some threshold V_i, and be quiescent otherwise. To take stochastic noise effects into account, the above deterministic rule may be generalized by requiring neuron i to fire with some probability depending on its local field. In analogy to thermodynamics, the first, deterministic rule can be considered as a Monte Carlo dynamics at zero temperature (T=0), the stochastic version operates at finite temperatures (T>0). One should keep in mind that the term "temperature" in our context always refers to the degree of disorder in the replies of the individual neurons; it should never be mixed up with the ordinary notion of temperature.

To describe the actual state of the system it is convenient to introduce another type of variable: the overlap m_μ measuring the correlation between the actual state and the stored pattern μ. Expressed in a slightly different way, m_μ tells to what extent pattern μ is matched by the actual state. If m_μ=1, the system is in pattern μ. However, if m_μ vanishes, there is no relation whatsoever between the actual state and pattern μ. So, this order parameter is the natural quantity which is needed to describe the behaviour of the network in terms of the stored patterns.

Now back to the action. For the moment we will restrict ourselves to the deterministic case of retrieval (T=0). Here the stored patterns are required to be stable fixed points of the network's dynamics. An external stimulus which sets the network to an initial configuration causes one of two typical behaviours. If the starting configuration is similar to one of the stored patterns, say μ, and the initial overlap m_μ is large enough (>0.5), then the system quickly settles down to that pattern. Otherwise, it will wander significantly longer before some final state - not necessarily one of the stored patterns - is attained.

In less formal terms, the above setup may be visualized as the construction of a landscape over the state space (the possible firing patterns) of the neural net. Each location in this landscape is associated with a firing pattern of the system and the dynamics described above corresponds to a downhill motion in this landscape. By shaping the landscape (through learning) such that its valleys are associated with stored information, the system functions as an associative memory: An initial state which "somehow resembles" one of the stored prototypes corresponds to a location in the landscape which is close enough to the valley representing that prototype to lie in its basin of attraction. By spontaneously moving downhill, the network reconstructs the prototype.

The landscape metaphor constitutes in fact more than just a convenient visualization of the processes involved in learning and information retrieval. For networks with symmetric couplings, it can be shown to have a rigorous mathematical counterpart.



Picture reconstruction by a neural network

Following its internal dynamics, a neural network may start with a new stimulus and gradually evolve to one of the pictures which were used to mould its memory.

For this illustration, the program "Spinning Brain" was first trained on the pictures of the album given around. Then the rudimentary pictures far outside were presented to the net. The rest shows what the dynamics of the program led to.

Fig. 2: Picture reconstruction by a neural network

To summarize the basic features: The external stimulus (e. g., a question) directly determines the final state (the answer) and the memory is content-addressable. The mechanism even allows retrieval of input data which do not match exactly any of the stored prototypes ($0.5 < m_\mu < 1$). One might compare this to recognizing a person on a blurred photograph.

b. Mixtures, spin-glass states, and storage capacity

Alas, the stored patterns are not the only attractors of the retrieval dynamics. There are quite a few other ones, the so-called spurious states, which are *not* wanted. In the landscape picture, these are "side valleys" which are unavoidably created along with those which are desired. As long as the number *q* of stored patterns is not too large, the spurious states represent just *mixtures* of a relatively small number of patterns. The number of mixture states increases exponentially with *q* but, luckily, one can get rid of them by raising the temperature. The explanation is simple: Mixtures are not as stable as a single pattern states - the corresponding valleys being rather shallow - so that the system can escape from them through "thermal" agitation. As the temperature T is raised further, "thermal" agitation becoming more violent, eventually a well-defined critical temperature T_c is reached above which the system also escapes from deep valleys, so that even the retrieval of the patterns themselves becomes impossible.

As the number *q* of stored patterns becomes bigger and bigger, the mixture states, whose number also increases, begin to merge and intermingle in an increasingly complicated fashion until eventually they swamp the whole state space - including the retrieval states which correspond to the stored patterns - to form a highly complex landscape corresponding to a so-called *spin-glass* state. The name refers to spin glasses, substitutional alloys with magnetic moments (spins) whose complicated dynamic behaviour was only partially clarified recently. It has turned out that the methods developed for the spin-glass problem apply remarkably well to collective phenomena occurring in the Hopfield model, such as recollection of patterns.

According to Hopfield, spurious states might have some importance to the development of creativity. Both mixtures and spin-glass states are derived from the existing patterns but the combinations - if you wish, associations - may be quite unusual. This implies that total suppression of spurious states might not be a desirable goal - except when you fear people's creativity. On the other hand, in case the spurious states are not suppressed at all, the creature might be unable to survive because of its hyper-creativity; the imaginations might prevent adequate reactions to environmental conditions. The reader is cautioned that the foregoing is, of course, rather speculative.

The next question we want to investigate is: How many patterns as compared to the number N of neurons can eventually be stored? Analytical and numerical calculations show that the storage level increases linearly with N. The critical proportionality constant α_c is called the *storage capacity* and is about 0.14 for Hopfield networks. Up to $\alpha_c N$ patterns are stored and retrieved almost perfectly (the error is less than 1.5%). If the maximum storage is attained and additional patterns are stored, the memory gets to a state of total confusion. *No pattern can be recalled any more.* This is quite an unattractive feature and should be repaired (see below: § 5.a, 'Forgetful memories').

c. Idealizations and their justification

The Hopfield model embodies a number of idealizations, which stand in contrast to physiological data, but which were crucial in taking the first step to get systematic and analytical investigations off the ground - namely in discovering some of the simpler system theoretical questions that lay hidden in the gamut of neurophysiological data. The influence of these idealizations on the validity of the modeling is not clear a priori. To estimate it, computer simulations have turned out to be instrumental.

The first presupposition a neurophysiologist would worry about is the employment of *symmetric* synaptic efficacies. In general, if some neuron transmits signals to another neuron with a certain intensity nothing guarantees that the transmission in the opposite direction - if it exists at all - happens with the same strength. Predominantly, in nature, it won't. If the bonds of a Hopfield network are removed randomly and asymmetrically (J_{ij} is set to zero and J_{ji} keeps its value), the new network behaves quite similarly to the original one, even at high degrees of dilution. The main change is a slight reduction of the storage capacity. A similar robustness occurs if we clip the J_{ij}. *Clipping* means that instead of the full-blown J_{ij} we only take their *sign*. This implies a considerable reduction of information that is actually imprinted in the network. Nevertheless, the storage capacity and retrieval quality are hardly reduced.

The next unrealistic feature, however, is the *high mean activity* of the network: On the average fifty percent of the neurons fire. In the brain, the fraction of firing neurons is about ten percent. It is possible to drop the simplifying assumption of high neuronal activity at the cost of getting large correlations between the patterns stored in a so-called low-activity network. Meanwhile one has succeeded to deal with these correlated patterns without leaving the framework of simple, local synaptic efficacies too far.

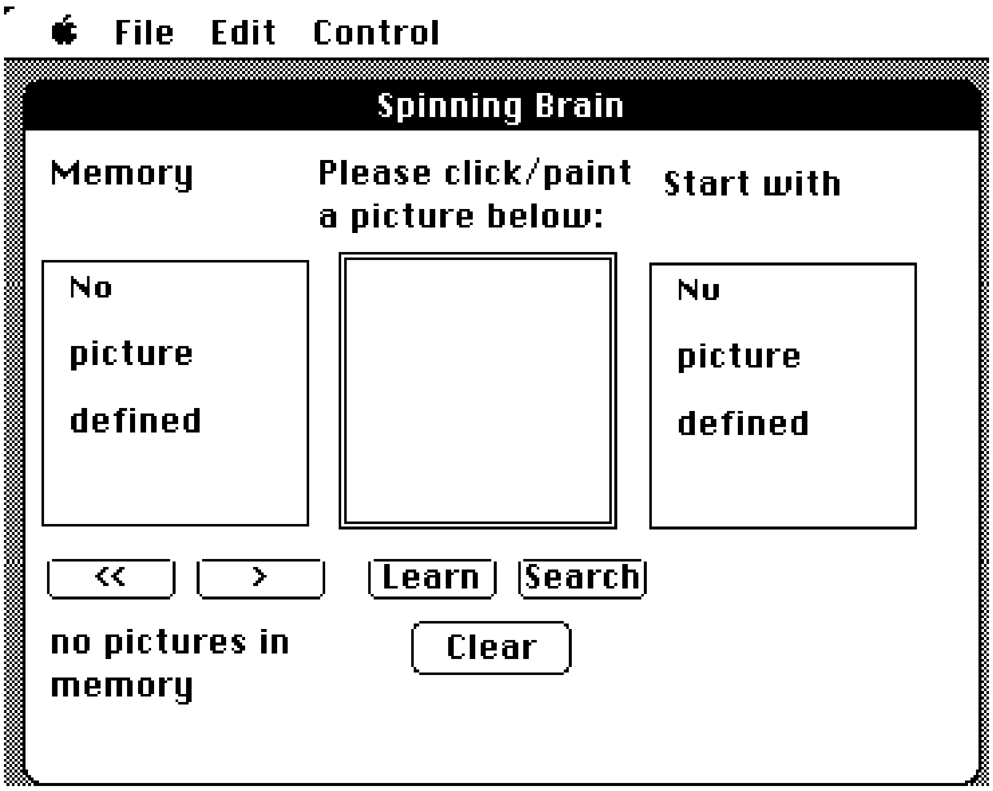
The last point worth mentioning is the treatment of *neurons as two-valued entities*. The inner structure of action potentials (spikes) and spike trains is regarded to have no influence on the *global* operation of the network. This assumption is justified by two facts. First, spikes are almost equal in shape and intensity. Second, if one looks at recordings of an individual neuron's activity, one observes only two different types of behaviour. Either low activity with a few spikes randomly fired over a period of tens of milliseconds, or maximum activity with a lot of equidistant spikes, separated by only a few milliseconds. We should however not fail to mention that, despite of this, eminent scientists do hold the opposite view, namely that individual spikes *are* relevant for the global dynamics of the network.

4. Spinning brain: Program architecture and performance

Up to now, patterns were assumed to be random. For analytic purposes this assumption is quite convenient but for practical work it is by no means necessary and, what is more important, not generally the case. Imagine, for instance, that you draw with your mouse several trees, animals, and other objects. These patterns then have to be stored in the synaptic efficacies in such a way that they do not get mixed up. In the *Spinning Brain*[3] this is accomplished by using a quasi-inverse technique which in its simplest form dates back to work of Kohonen (1978) in the seventies.

In the *Spinning Brain* program, we use a neural networks algorithm to recover a picture from a collection kept in memory. To see what is happening, we keep an album of the stored pictures. The use of pictures is slightly misleading: actually our neural net has no geometrical knowledge. It does not know about directions or straight lines or distances - it is just modelled as a collection of neurons. For moulding the memory during learning, it only is noted whether two neurons, while the network represents an item, fire at the same time - be they adjacent or far apart. Each pixel of the picture is associated to one neuron. In this simplified model, it only matters whether two pictures overlap. For a more realistic model one would like to include other features which are so important to our visual system: we do not perceive the world as frozen pictures, but we see in a dynamic way. Our eyes constantly move, the focus is ever changing, adding to our geometric perception of the world. For illustrating neural network theories, we found the current reduced model sufficient. We are however aware that our model will not be able to detect the similarity of two lines of same length, one of them being only shifted to another position.

[3]3 The name alludes to the Ising spins used to represent the data.



Memory

Please click/paint a picture below:

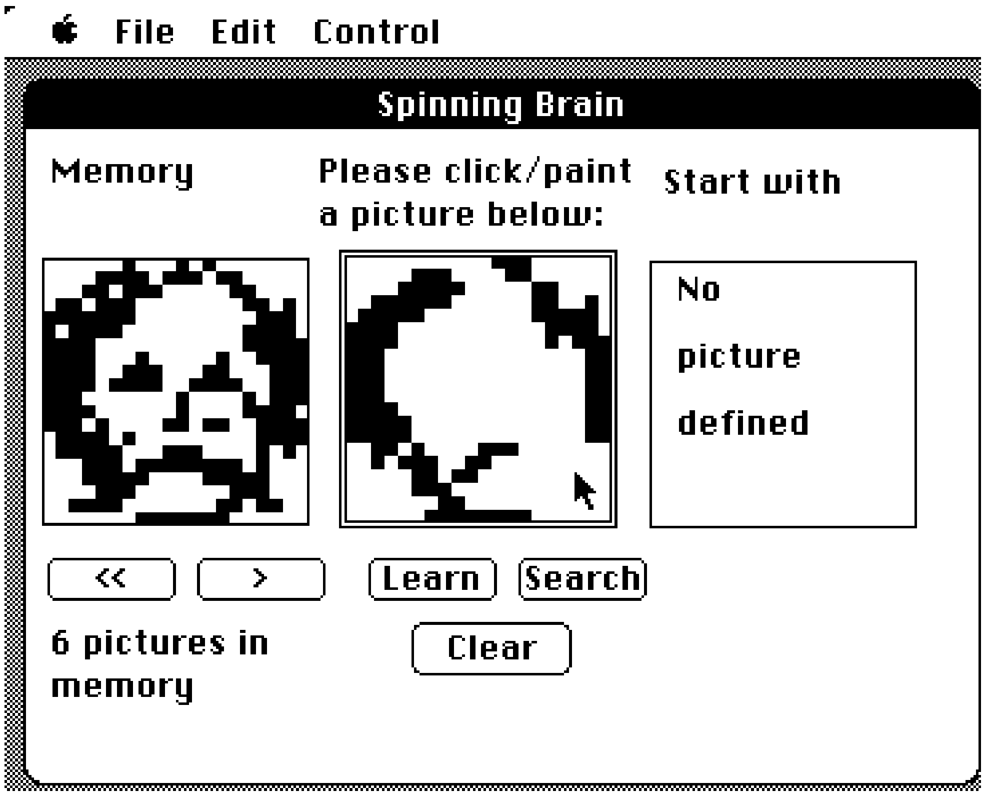
Start with

No picture defined

No picture defined

no pictures in memory

Fig.3: Initial window of "Spinning Brain". With the mouse, you can enter a picture in the center frame as fat pixels, or you can read an album from disk. "Learn" will add the new picture to the album, "Search" will activate the neural net.



Memory

Please click/paint a picture below:

Start with

No picture defined

6 pictures in memory

Fig.4: Entering a picture in "Spinning Brain". You can click or draw a picture with the mouse.

In formal terms the *Spinning Brain* program is organized as follows:

INPUT MODE:

- A *pattern* $\{\xi_{i\mu}, 1 \le i \le N\}$ is presented to the neural net, where $\xi_{i\mu} = \pm 1$, if in pattern μ the i'th neuron is firing/quiescent.

LEARNING MODE: (called upon going from input mode to retrieval mode)

- Compute the *correlation matrix* C with elements

$$C_{\mu\nu} = N^{-1} \sum_{i=1}^{N} \xi_{i\mu}\, \xi_{i\nu} \qquad , \ 1 \le \mu,\nu \le q'$$

where *q* denotes the total number of patterns in store.

- Invert C (*e.g.* by the Gauss Jordan algorithm) :

$$\underline{C} := C^{-1}$$

- The synaptic efficacies J_{ij} are then given by

$$J_{ij} = N^{-1} \sum_{\mu,\nu} \xi_{i\mu}\, \underline{C}_{\mu\nu}\, \xi_{j\nu} \qquad , \ i \ne j \ ,$$

$J_{ii} = 0$.

RETRIEVAL MODE:

- Starting from an initial firing pattern $\{S_i\}$ (a picture to be recognized), compute the postsynaptic potentials (PSP's)

$$h_i = \sum_{j=1}^{N} J_{ij}\, S_j \qquad , \ i = 1,\ldots,N \,.$$

- { Monte-Carlo Algorithm: }

```
Repeat until a stable state is reached
    Select i ∈ {1,…, N} at random
        Update
          If  $h_i S_i \ge 0$ goto Select          { do not change state of neuron i }
          If $h_i S_i < 0$ then
            $S_i \rightarrow -S_i$                          { change state of neuron i }
              $h_j \rightarrow h_j + 2 J_{ji} S_i$ ,  j = 1,...,N.     { and update PSP's }
        End {Update}
    End {Select}
End {Repeat}
```

{A stable state is defined by the condition that $h_i S_i \ge 0$ for *all* i. In *Spinning Brain,*the Monte-Carlo algorithm stops, if in a certain number (approximately 1.2 N) of successive attempts no neuron changes its state. Since in the Monte-Carlo dynamics the neurons are selected at random, true stability is thus only guaranteed with (sufficiently) high probability.}

The program runs essentially in three phases: first, the album of the *Spinning Brain* is loaded with a series of pictures. The user can click these pictures with the mouse as fat pixels. Then we "give the brain a look at these pictures". In mathematical terms, we compute the synaptic efficacies J_{ij} so that the pictures of the album become stable attractors of the neuro-dynamics. Once this is accomplished, the user can enter another picture using the mouse. The program will then try to fit the new picture into memory. That is, starting from the new picture and following the neuro-dynamics, it evolves through a sequence of network states (other pictures), until it eventually ends up in a stable state, (hopefully) showing one of the pictures which were used in forming the memory. But it can also be mislead, ending up in a chimera, corresponding to one of the spurious states mentioned above.

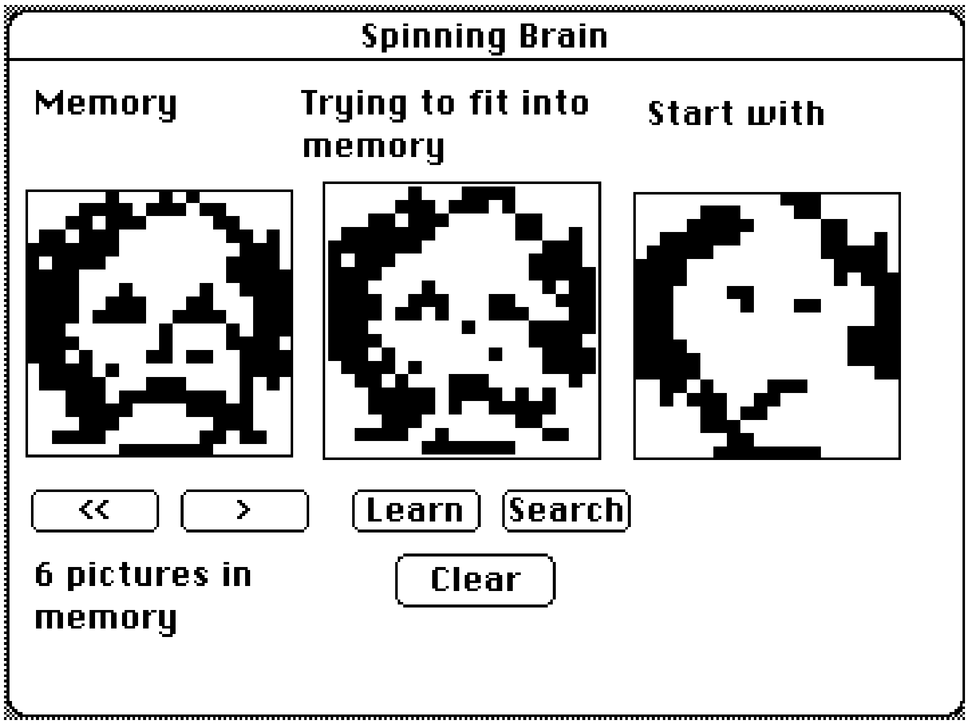


Fig.5: "Spinning Brain" in action. In the center frame, you can follow how the picture to the right is being fitted into memory.

Since the reconstruction can only take place when the dynamics has been fixed, the moulding of the dynamics takes place automatically if the user asks to fit a new picture into memory and before the search actually starts. So to the user only two modes are visible: adding pictures to the memory, and retrieving a picture from memory.

To allow for quick program development, the program makes strong use of the facilities of the Macintosh operating system and leans heavily on a programming framework. We used *Random&Template*, the programming template of the Heidelberg StatLab[4]. *Random&Template* is an extensible Pascal program, written in "classical" Pascal, but introducing many concepts oriented towards Object Pascal and MacApp. *Random&Template* has been introduced to teach program design and to smooth the transition to Object Pascal. It provides all the generic event handling. Spinning Brain was build on top of *Random&Template* by overwriting the facility for handling modeless dialogs as provided by *Random&Template*.

[4] *Random&Template* is in the public domain. The most recent version can be ordered at DM 29,- for disks and handling from StatLab, Im Neuenheimer Feld 294, D 6900 Heidelberg.

5. Extensions: On forgetting, dreaming, hierarchies, and singing

a. Forgetful memories

We have seen that as soon as the number of stored patterns exceeds the maximum storage capacity, no pattern retrieval is possible any more. This does not seem very realistic. The point is that, as the storage level increases and approaches the critical value, one would like the memory to forget some of the old patterns in order to preserve room for the new ones. This can be achieved by two mechanisms, which differ radically, and which are not generally agreed upon. One may forget because certain rather complicated molecules are degraded as time proceeds. This is chemistry and may be modeled by giving the patterns a weight which decreases with storage ancestry. Alternatively, forgetting is an *intrinsic* property of the network. This is modeled rather naturally by putting bounds on the synaptic efficacies as follows. When a new pattern is stored, one has to add or subtract 1/N (cf. above, § 3.a) to or from the former value of the couplings, provided it does not yield a result which exceeds in absolute value a certain bound. The recently learned patterns now remain retrievable - the memory forgets the old ones.

b. Dreams

According to Crick and Mitchison, dream (REM)[5] sleep improves the performance of a neural network in that, after a bunch of dreams, it remembers the memorized patterns much better than before. To model this phenomenon, one starts with a Hopfield network which has stored a certain number of patterns. This number is fixed throughout what follows. Now, the retrieval process is started from a random initial configuration. Under its dynamics the system relaxes to some final state (cf. § 3.a). This procedure is called a "dream". Then the bonds are modified: A term d/N is subtracted from the previous J_{ij} if in the dream's final configuration the neurons i and j are in the same state; otherwise d/N is added. d is a dream-factor, weighting the influence of a single dream. This procedure is repeated. After a number of dreams, one observes that the basins of attraction (i. e., the sets of configurations which tend to a particular stored pattern under the system's dynamics) are enlarged and the retrieval quality greatly improved. The underlying mechanism is not yet fully understood and still under investigation.

c. Hierarchies

We are accustomed to use fundamental concepts which constitute categories in our way of thinking and less fundamental concepts which classify objects within a particular category. For example, talking about buildings in general and about a hut, a palace, or a skyscraper in particular. There are different approaches to the construction of neural networks which can handle their percepts with different degrees of generality. Some have a tree-like "hardware" structure, each branch being responsible for a particular degree of generality, some realize that task in an architecturally undifferentiated network with a tree-like "software" structure. Both types of model share the following operational feature. In a first step the percept is attached to a category on the fundamental stage by a coarse classification. Next, it is compared to the items within this category and attached to objects which are "similar" to it; and so on. Thus the perception is guided by a hierarchical organization which in turn is modified by the learning process. Retrieval on a particular hierarchical level in principle works as in an ordinary Hopfield network.

[5] REM refers to the *r*apid *e*ye *m*ovements that accompany such phases of dream sleep.

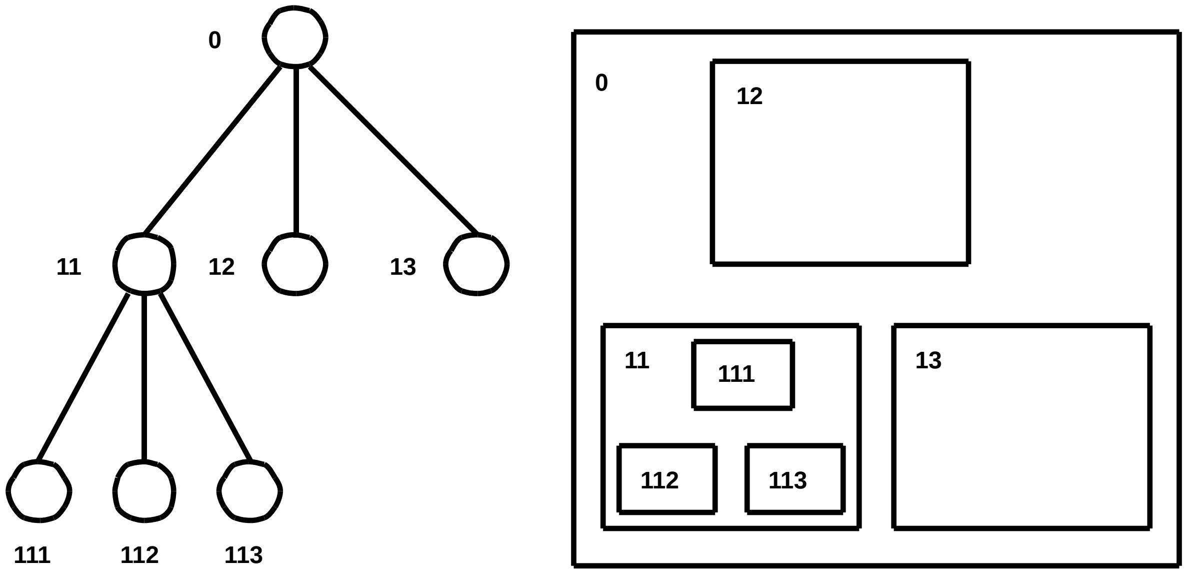


Fig.6: Two views on hierarchical data organization. One is in terms of a tree-like structure, the other utilizes a system of boxes within boxes. For Apple users, the latter view is familiar from the hierarchical file system (HFS).

d. Temporal sequences

Networks with symmetric synapses, as considered above, only allow for the storage of static, time independent patterns. But we wish (don't we?) to learn temporal sequences of patterns as well, e. g., singing a song, dancing, and other activities of that kind. Spin-glass-like neural networks can be designed to do that job. A successful approach was to *add* an asymmetric term to the original symmetric Hopfield bonds which is designed to induce transitions from one pattern to some other, and to introduce a delay for the transmission of the signals which are sampled via the transition term (i.e., $h_i=h_i^{Hop}+h_i^T$, where $h_i^T = \sum_j J_{ij}^T \overline{S}_j$ is a transition term and $\overline{S}_j$ a retarded signal from neuron j). The delay is necessary to stabilize the network in a pattern before the transition to a next pattern occurs. It also determines how long the system remains in a specific pattern. If the delay is missing or too short the network soon gets mixed up and exhibits irregular behaviour.

At first sight, the introduction of asymmetry and time delays in the synaptic efficacies in order to create transitions seems to be fairly ad hoc. A refinement of the Hebb rule, however, has naturally lead us to conceive bonds of the above mentioned structure permitting the storage and retrieval of both static and dynamic objects within a single network.

The basic idea is inspired by the observation that signals in the brain travel at a finite speed so that between different pairs of neurons they are transmitted with different delays. The learning rule for Hopfield networks demands that the synaptic efficacy be enhanced, if the neurons connected by this synapse are in the same state simultaneously. Now, if one takes into consideration the existence of delays τ_{ij} between neurons i and j, simultaneity *at the synapse* (i-j) means that neuron j was in the same state as neuron i a time τ_{ij} earlier. If a *broad* distribution of delays is realized in the network, the various delay lines serve to record *different* aspects of the temporal structure of a pattern sequence presented to the network. If the sequence is such that any pattern is present for a period significantly longer than the longest delay, the sequence is perceived as a set of independent static patterns. If the duration of the patterns is within the range of delays the signal will be perceived as dynamic pattern sequence. As the range of delays grows, the variety of representable objects increases.

6. Discussion

In the present context we have represented neurons as simple switches (flip-flops) embedded in a fully or highly interconnected network. Using this simplified setup we have seen that the recent input of physics in brain theory has provided two new elements: (a) the analysis and interpretation of certain brain functions as *collective* phenomena and (b) the notion of temperature to take care of internal noise arising from the system's activity. In this way a satisfying explanation has been obtained of the brain's surprising fault tolerance with respect to both input data errors and internal failures. Furthermore, the discrepancy between the simplicity of the very many neurons and the richness of behaviour of the composite system has been resolved.

One has to realize, however, that the above modeling is only a crude approximation to reality. If the aim is to model higher brain functions, one has to take into account the inner structure of the brain, including preprocessing of the stimuli. Yet, if one cannot understand relatively simple structures which, as we have discovered, already exhibit a surprisingly intricate behaviour, any further modeling is moot.

We have also seen that one can dispense with theoretical simplifications and that sensible programming using "neural" principles even manage the storage and associative retrieval of visual patterns as they occur in everyday life. The authors do not think, though, that intelligent human reasoning will be replaced by the physics of neural networks within a fair amount of time - if ever.

References:

Mc Culloch W.S. and Pitts W. (1943), A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. **5**, 115-133.

Domany E. (1988), Neural Networks: A Biased Overview, J. Stat. Phys **51**, 743 (1988)

Hebb D. O. (1949), *The organization of behavior*, (Wiley, New York).

van Hemmen J.L. and Morgenstern I. (eds.) (1986), *Heidelberg Colloquium on Glassy Dynamics,* (Springer, Berlin, Heidelberg, New York)

Hopfield J.J. (1982), Neural networks and physical systems with emergent collective computational capabilities, Proc. Natl. Acad. Sci. USA **79**, 2554-2558.

Kohonen T. (1978), *Associative Memory - A System Theoretical approach,* (Springer, Berlin, Heidelberg, New York).