Describing Computation using Attractor Superdynamics

Gene Chalfant
April, 1990

Foreword

November 12, 2024

When I wrote this, about 34 years ago now, a new concept called *connectionism* was spreading quickly in the academic universe. Simply put, this is the idea that artificial intelligence could be achieved not by writing programs, but rather by building a brain-like model with a tremendous number of nodes and connections. We knew this was what the brain looked like when we looked at it through a microscope, and we knew electrical pulses sent information down a nerve between the brain's cells. Ideas based on how this architecture resulted in a thinking human came from the cognitive biologists who built models and called them artificial neural networks. Tiny models of a few neurons and nerves were being built as early as the 1940s that showed intriguing behaviors.

In 1969, a couple of influential computer scientists wrote a book saying this could not possibly work. All funding through government agencies such as NASA halted, which led to the computer industry also being uninterested. The connectionists got neither national attention nor big grants; instead, they were ignored or ridiculed by the mainstream. This began one of the largest ice ages in artificial intelligence research.

When I wrote this in 1990, I was an enthusiastic PhD candidate who had studied the mainstream artificial intelligence efforts in great depth. I was much more interested in the work of neuroscientists on animal brains and why these models didn't agree. A few rebels insisted that, to build a machine that could think, connectionism was the only way forward.

I worked at NASA at the time and my project managers were notably unimpressed with these new ideas. They preferred that I build an incrementally faster computer with incrementally smarter programs, safe and sure to succeed. Proposing to build a new artificial brain to power a robot put you in a fringe category of alternative researchers considered irresponsible and lacking in gravitas. I was this kind of researcher.

Fast forward to 2024, and connectionism is ruling the largest tech companies and is well on its way to conquering the world, literally as well as figuratively. It turns out that copying the brain is a good approach. Connectionism needed a refocus on training rather than explicit design of behavior, structuring layers and pathways into an engineered learning machine, and most of all, huge arrays of much faster and cheaper computers. It also required lots of people and encouragement (funds) to spend their lives working all this out.

This paper proposes dynamical systems (math) descriptions of knowledge and intelligent behavior using attractors. It does not say much about how you might actually *build* something like this. We now know that such a network should be trained rather than built, and that you need a vast amount of compute, far more than was available in the biggest supercomputers at the time. Yet it still (I hope) captures some of the excitement in the artificial intelligence world as the big freeze ended.

Describing Computation using Attractor Superdynamics

Eugene Chalfant

April 1990

Abstract

Much work has been done recently on nonlinear dynamical systems of differential equations. The behavior of these systems often closely resembles that of real-world phenomena. Although nonlinear differential equations generally cannot be solved analytically, a complex system's dynamics can be described succinctly using attractors in a space of many (potentially infinitely many) dimensions. In addition, the dynamics of the attractor itself (i.e., how the system's behavior changes as a function of its environment), called *superdynamics*, may be described as a trajectory through the superspace of all possible attractors. This paper proposes a connectionist description of knowledge and theory of computation based on attractors and superdynamics.

Introduction

Since the early days of artificial intelligence, there have been two camps of thought on how to design an intelligent system. The mainstream of AI research describes knowledge on a conceptual level, essentially modeling high level processes according to logic. In this camp we find expert systems, scripts and schemas, and rule based systems. Connectionists, on the other hand, use complex networks of simple processors to try to implement intelligent systems at a very low level, modeled after the brains of living things. These researchers have developed backpropagation networks, associative memories, and other constructs which have certain properties in common with intelligent systems.

Both approaches have had limited success. The mainstream approach suffers from a lack of subtlety and elegance – many fundamental properties of intelligent systems must be handled by add-on mechanisms, such as learning, or dealing with novel input in an open system. For instance, refining the behavior of an expert system to handle special circumstances usually means adding special case rules to the rule base.

Published in Bernardo Huberman's *Virtual Journal*, Xerox Palo Alto Research Center, 1990

The connectionist approach has its own set of completely different problems. Many properties not displayed by high-level AI systems are emergent properties of connectionist systems, such as generalization, categorization, and associative recall. The overwhelming complexity of the connection topology of neural networks and the difficulty of examining their internal distributed knowledge representations make engineering these networks a hit-or-miss proposition.

What is needed is a description of knowledge and computation which covers both camps, is scalable from low level networks to high level complex systems, and which explains the shortcomings of either approach. With scalability comes the ability to bridge the rift between the two approaches, opening the door to cross-pollination of ideas and techniques. This paper proposes such a description based on a relatively new branch of mathematics which has already shown promise in modeling physical processes in nature.

A brief introduction to attractors and nonlinear dynamics is presented, followed by a description of how these concepts can be related to information theory. Next, a connectionist implementation of a semantic network is described. A translation of this implementation into attractor terms is then presented. The paper concludes with a speculative discussion on the application of the attractor paradigm to several real-world problems.

Attractors describe the behavior of complex systems

What is an attractor?

- **Each possible pattern of activations distributed across a network collapses to a single point in state space**

Imagine a complex network of processing nodes or neurons. Each node has a certain activation, which may be binary or real-valued, depending on the model being used. If we take a snapshot of all activations across the entire network, we can describe the state of the entire network as a single point in *n*-dimensional state space, where *n* is the number of nodes in the network. Thus the distributed pattern of activations at a single instant in time collapses to a single point in state space. The state space contains *all* possible states of the network. If the values which a node activation may take are constrained, the state space is similarly constrained.

- **Patterns of activation generally change over time**

Now, rather than taking a single snapshot, we take a sequence of snapshots to form a movie of the pattern of activations as the pattern changes over time. In state space, we now get a trajectory in state space which describes the exact behavior of the system over time.

As an example of a dynamical system, imagine a frictionless pendulum in motion. Plotting its behavior in a two-dimensional state space where kinetic energy (speed) and potential energy (height) are the two dimensions results in a circular state trajectory. The current state of the system is a point which revolves forever around this circle.

- **Trajectories in state space are "attracted" to certain points which correspond to temporally stable patterns**

A nonlinear dynamical system, in general, has certain states which minimize the "energy" (or identically, maximize the entropy) of the entire system. The system has a tendency to fall into these minimum energy states. Whatever the current state of the system, its state will change over time until it reaches one of these minimum energy states. Once it enters one of these states, it stays there until the system is externally influenced. These stable states of the system are called *fixed point attractors*.

An attractor describes all possible behaviors or state trajectories of a dynamical system, given an initial system state. In state space, the attractor is an *n*-dimensional point, and trajectories describing the evolution of system state are "attracted" to that point. There are an infinite number of possible trajectories. When the system state settles toward an attractor as a function of time (as opposed to other external influences), it is called a *temporal* attractor. The attractors described in this paper are temporal attractors.

If we re-examine the pendulum described above, we may form a fixed point attractor by adding friction to the system. Friction causes the pendulum to eventually end up in a zero energy state, corresponding to a point at the origin of the state space. This point is the attractor describing the system. No matter what the starting state, the current state will spiral into the zero energy point as the energy of the system is dissipated by friction.

There are three fundamental types of attractors: Fixed point, periodic, and chaotic. When the trajectory settles to a repeating cycle of patterns, the attractor is called a *periodic* or limit cycle attractor. This type of attractor is characterized by a trajectory which exactly repeats itself (i.e., the system periodically returns to *exactly* the same state it was in previously).

A *chaotic* attractor has an underlying pattern, but it appears as though the system behaves randomly. For example, weather has been described by a chaotic attractor (the Lorenz attractor). It behaves somewhat repetitively in cycles of days and years, but its exact behavior cannot be predicted. One can at best estimate the probable range of weather for a particular season and time of day. In state space terms, we can estimate a bounding hypervolume but we cannot determine future states (i.e., the state trajectory) exactly.

The more complex periodic and chaotic attractors can have an extremely complicated structure when visualized in three-dimensional space. As state space dimensionality increases, so does the number of possible attractor shapes. Also, an attractor may appear periodic until magnified, when chaotic characteristics emerge.

- **An attractor describes all possible trajectories in the state space of a given system whose initial state is within its "basin of attraction"**

One can think of the state space as having an "energy landscape" where attractors are the low points, and the strength of the attractor is determined by the gradient of the landscape. This energy landscape completely describes the system's dynamics. If the system can be described with a single attractor, the entire state space is within the

attractor's *basin of attraction*. However, a system may have more than one attractor and therefore more than one basin of attraction. In the same way that watersheds determine where a raindrop will end up, a basin of attraction indicates that any initial condition within the basin's boundaries means that the system state will eventually end up at the associated attractor.

- **Attractors change as a function of the control parameters of the system**

So far, we have discussed attractors as static entities, unchanging over time, even though these static attractors describe the dynamic behavior of the system. However, the attractors themselves may change for a number of reasons. The factors which cause an attractor to change (thus changing the dynamic behavior of the system) are called *control parameters* of the system. Control parameters are essentially anything that can change the attractor that describes the system. Control parameters of a system can include external environmental inputs (attractors change because of new input to the system), time decay dynamics (attractors change over time), global parameters, or anything else which may affect overall dynamical behavior. For example, one control parameter for the weather attractor is insolation (amount of incoming solar radiation), which is an external input. Another control parameter for the weather system might be the level of production of manmade air pollution.

The description of the way attractors change as a function of control parameters is called *attractor Superdynamics*. A discrete change in an attractor (implying a change of behavior of the system) is called an *attractor transition*. The notion of attractor transitions implies a discrete model in which a stable attractor transitions through intermediate unstable attractors into another stable attractor.

- **Superdynamical space is a yet higher level description of dynamical systems**

Consider a space in which each point describes the behavior of a dynamical system – each point then represents an attractor. The dimensions of this space are control parameters. We are collapsing the entire state space dynamical description into a single point, just like we collapsed the pattern of activations across the network into a single point in state space. This (potentially) infinite dimensional space of all possible attractors where each dimension describes a control parameter is called *superdynamical space*.

Control parameters are not necessarily independent. A control parameter may have a positive correlation with another, which geometrically means that the two corresponding dimensions are no longer orthogonal. A full correlation indicates that the two dimensions have essentially rotated so that they are collinear.

A *superdynamical scheme* is a "trajectory" in superdynamical space which describes the way an attractor changes in a system as a function of control parameters (Abraham/Shaw). A superdynamical scheme describes a continuously changing attractor, as opposed to a sequence of discrete attractor transitions.

What is the rationale for using attractors to describe computation?

- **Systems of nonlinear differential equations describe the hidden regularities inherent in many natural phenomena**

Until the advent of computers, systems of nonlinear differential equations have been intractable problems, since they generally cannot be solved analytically. The computer allows the behavior of these systems to be examined numerically. This research has revealed that solutions of even simple systems can have intricate structure (e.g., the Mandelbrot set). These systems can also generate lifelike physical structures, such as fern leaves, clouds, etc., by modeling actual physical processes in nature, which tend to be nonlinear. These systems seem to contain essential regularities of nature based on physical laws.

- **A basic assumption in this paper is that certain aspects of information about the world have a similar structure to the world itself**

Why has intelligence evolved? According to Darwin, survival is the prime motive behind any successful evolutionary trend. Intelligence enhances survival by enabling the organism to predict its environment – where food can be found, how to avoid becoming food for another. To predict its environment, an organism must have an internal model. This internal model is used to recognize and recall regularities in the organism's environment. This paper discusses the possibility of using nonlinear dynamical systems as an internal model.

- **Acquisition of knowledge as the constrainment of possibilities**

The process of acquiring knowledge can be thought of as a progressive containment of possibilities. One way of constraining possibilities is by extracting properties which are found to be important in defining a concept, and giving these properties values. A pair of concepts originally indistinct from each other would thus be separated on the basis of some observed property. For example, before one learns about the concepts *blueberry* and *strawberry*, their colors are unknown. Upon learning that strawberries are red and blueberries are not, the property "redness" may be used to indicate that if redness is absent, you're definitely not looking at a strawberry, but you still might be looking at a blueberry. The fact that strawberries are red has enabled a partial classification or partitioning of the world into red-things and non-red-things, with strawberries in the red-things partition and blueberries in the larger non-red-things partition. The concept "strawberry" has been constrained to exist in the red-things partition only. Thus the range of possibilities has been limited in the "redness" dimension. One may think of this as a physical constrainment along a single dimension in an infinite dimensional concept space.

The game "Twenty Questions" consists of a process of constraining a domain of all possible answers down to a very specific concept by successive partitioning of possibilities. In this game, one may think of an infinite dimensional hyperspace of possible answers. When the questioner chooses a question, he or she is essentially partitioning this

hyperspace along a single (binary) dimension. If a question is "Is the object red?", the binary yes-no answer limits the space of possibilities. Skillfully chosen questions can very quickly reduce a universe of possibilities to a single one; this skill is the mark of a good player.

This approach works naturally with fuzzy answers as well. An object one inch in diameter may or may not be a strawberry. If it is four inches in diameter, it is very unlikely (but not impossible) to be a strawberry. While the concept "strawberry-ish diameter" exists as a fuzzy probability distribution on a dimension, an observed instance of some object and its diameter can indicate the strawberry-ness of that object in that dimension.

A berry botanist, as a trained observer of properties, will be able to use many more dimensions to classify a strawberry than a layman, and can classify the strawberry into varieties unseen by anyone else. A dessert chef might classify the same strawberry into completely different categories such as taste, texture, and fragility. Nevertheless, all possible strawberry varieties and variations should still be in the same vicinity of the concept hyperspace.

- **Attractors and connectionist networks are both low-level descriptions of complex systems which can be used to describe, recognize, and manipulate symbols**

In this paper, we seek a method whereby the low-level structure and behavior of a complex system may be abstracted to higher conceptual levels which can describe, recognize, and manipulate symbols, thereby bridging the gap between mainstream AI and connectionism. The major problem is the extreme complexity of these low-level systems.

The method proposed here uses an equally extreme simplification and abstraction of low-level dynamic behavior using attractors and superdynamics, keeping only the essence of the behavioral description. We must also radically simplify this representation to keep it comprehensible as we can only visualize a two or three dimensional subspace at one time. However, once the basic concepts are established, it is not difficult to imagine how a number of these subspaces can work together as a concept discrimination mechanism.

Attractors describe the dynamics of complex, non-linear, real-world systems

- **Attractors exist in a highly dimensioned state space of a complex system**

Attractors are a mathematical construct which exist in a high dimensional (potentially infinite-dimensional) continuous state space of infinite extent. This state space can completely describe a physical system, such as a neural network, using a finite dimensional subspace. The energy landscape metaphor allows us to visualize any two of these dimensions at a time as a hyper-cross-section, along with how the energy of the entire network changes as a function of only these two properties, while holding all other properties (dimensions) constant.

- **Neural networks are complex, non-linear dynamic systems with continuous input**

Consider the dynamics of some typical neural networks which have been implemented. The backpropagation network is usually a non-recurrent network (except during training cycles) in which network inputs propagate through a series of layers to the output nodes. If we hold fixed inputs, all nodes in the network will reach a stable value in a single pass. The network has thus settled immediately to a single, unmoving point in state space. No temporal dynamics are involved, so the system is statically describable and therefore does not have an attractor.

The more biologically plausible recurrent networks, on the other hand, are dynamical systems. These networks settle into a stable state after multiple passes, displaying dynamic behavior, and so these systems do have temporal attractors. Hopfield networks are an example of a fully interconnected recurrent network. Hopfield describes global network behavior using an energy landscape metaphor in a two dimensional state space. A fixed point attractor corresponds to a sink in this scheme. The state of the system can be described as a ball which rolls downhill along the landscape until the lowest point is reached.

- **A superdynamical scheme describes a particular progressive tilting or distortion of the energy landscape**

The energy landscape of a dynamic system such as a Hopfield network is fixed only as long as the attractors are static. In superdynamical space, this system's behavior is described by a single point – the control parameters, or inputs, are held constant. If inputs are changed, of the configuration of the network is altered (connection weights change) the energy landscape is also changed, along with its attractors. The point in superdynamical space then moves.

These changes to the network can be visualized as a tilting or distorting of the energy landscape. The effect of escaping local minima (or basins of attraction) is similar to that of simulated annealing in a Boltzmann machine. However, the mechanism of escape in a Boltzmann machine is the stochastic repositioning of the state (making the "ball" jump around) which diminishes over time, while varying control parameters cause the energy landscape to actually change in a meaningful way.

- **Boolean functions can be modeled by energy landscapes**

As a simple example, a system to solve the binary exclusive-OR function can be modeled using a three dimensional state space. The system should attain its minimum energy (be most "relaxed") when the correct answer is observed in conjunction with the current inputs. The classical XOR network consists of two input nodes, one output node, and a hidden node. We don't really care what the hidden node inside the black box does, so we examine the state space of the other three externally visible nodes. We represent state space coordinates as (input1, input2, output). Using only three dimensions for state precludes using one of them to represent state energy, so we need to think of energy as a

non-dimensional property of a state, like density, or better yet gravitation, which can have a gradient vector associated with it. Therefore, valid states of the network are at coordinates (0,0,0), (0,1,1), (1,0,1), and (1,1,0). Note that, since XOR is a discrete function – inputs and outputs are 0 or 1 – only the eight points at the corners of the unit cube are potential locations for the attractors. Intermediate states encountered while transitioning between stable states, however, might be real-valued. The energy landscape is then one with "gravitational" attractors at the points corresponding to these states. Clamping the inputs forces the network to eventually settle to one of these valid, stable states. The state of the system is "repelled" away from invalid states. Although any other Boolean function can be modeled similarly using the attractor paradigm, the XOR function is one that is nonlinear and thus requires a hidden node.

The Boolean function above places many limits on the attractor representation which are unnecessary. For example, there is no inherent reason to limit these functions to binary values. Fuzzy logical functions are a natural extension of the attractor representation, since the state space is continuous. Also, limiting the space to the unit cube is also an artifact of using binary values. This limit conceptually puts infinitely high walls on the (two dimensional) energy landscape, denoting illegal values. The attractor representation allows a designer great freedom in creating a landscape.

The example chosen here is not meant to suggest that biological systems use Boolean functions to implement computation, but rather to show the generality and illustrate the application of the technique.

Connectionist systems are also dynamical systems

- **The dynamical description is a higher level view of network behavior, but it is nevertheless a complete description**

Let us examine a higher level superdynamical description of network behavior. In using attractors to describe system behavior, we trade off a description of the actual behavior of the network as a single trajectory or sequence of states for a higher level description of all possible sequences of states and where they will end up. When we add attractor superdynamics as an even higher level description, we then get a description of how the system will change with changes in control parameters.

In doing this, we abstract farther and farther away from the "freeze-frame" description of a pattern of activations toward a description of the behavior of the system no matter what the current state, and eventually to the superdynamical description which predicts system behavior for every possible set of control parameters (external inputs). This is the same distinction as between the actual state description of a trajectory of a ball within a landscape, and the landscape itself which denotes all possible state trajectories. The superdynamical description further describes how *any* landscape will change as a function of control parameters.

This hierarchy of descriptions accounts for any external factors which can affect the system. It is fully extensible and fully scalable. It is, however, an exceedingly complex

representation which can only be comprehended by extreme simplification. This is to be expected from a highly complex system. The problem then becomes how to deal with this representation – how to relate it to currently used, more traditional representations, and how to physically implement it on a connectionist network.

Implementing attractor transitions in Hopfield-type nets: several methods have been proposed

- Using noisy and/or asymmetrical connections: (Buhmann & Schulten, 1988)

Buhrman and Schulten have proposed an implementation of attractor transitions which separates node connections into three types: (1) Symmetrical excitatory connections between units active in a particular pattern builds the strength of that pattern, (2) Inhibitory connections between active units in the pattern and all inactive units prevent two patterns from merging into one attractor, and (3) Temporal order is established by excitatory connections from units active in a pattern to units active in the next pattern in the sequence, and by inhibitory connections to active units in the previous pattern. The amount of global noise determines the stability of the attractors (i.e., the strength of the tendency to transition to the next attractor in a sequence).

- Using connections with built-in time delays: (Kleinfeld & Sompolinsky, 1988) and (Amit, Gutfreund, & Sompolinsky, 1985)

A technique using asymmetric connections with built-in time delays is proposed by several physicists. In addition to the symmetrical connections in the Hopfield net, effective instantaneously, asymmetrical connections which are not immediately effective serve to generate the next attractor (t+1) in a sequence. This subsequent attractor (t+1) then inhibits the units active in the current (t) pattern. The time delay allows the current attractor (t) to remain active for a certain amount of time by delaying the generation of the next attractor (t+1), thus delaying the inhibition of the current attractor (t). This model is particularly popular.

- Using activations which can decay over time (Schreter, 1988)

Schreter proposes a simple technique which overlaps units active in an attractor with the active units of the next attractor. While the current attractor is active, its active units spontaneously decay, removing the inhibition of other attractors. The next attractor is the one with the most active units which overlap the current attractor.

- Using connection asymmetries which can change over time (Peretto & Niez, 1986)

This technique also uses connections with both symmetrical and asymmetrical components. The asymmetrical component has temporal dynamics which effect the transitions to successive attractors. While the current attractor is active, the asymmetrical components increase with time until the next attractor is forced active.

- Environmental changes external to subnets can influence attractors (Bell, 1989)

Bell summarizes the preceding techniques in his 1989 paper, then presents a new technique in which attractor transitions arise from the influence of attractors in other nets. Units get input from other units in the same attractor net, which sustains the current

attractor. In addition, external inputs may or may not sustain the current attractor. Bell then suggests three basic ways an attractor may influence another attractor in a neighboring net: (1) sharing units, (2) feedforward connections from local active units to units in the neighboring net, and (3) feedforward connections from local active units to *connections* in the neighboring net. A model is presented which can store a limit cycle attractor or random transitions.

All these models provide valid mechanisms to transform from an attractor representation to a low-level network representation (activations and connection weights). These mechanisms suggest the potential of developing an automated technique whereby, once an attractor representation of knowledge is formulated, it can then be "imprinted" onto a physical network – in essence, a kind of atypical "learning" mechanism which is not trained by example, but rather engineered by design.

Knowledge can be encoded as constraints on superdynamical scheme "trajectories"

"Trajectory" as used here is not defined as a position change over time, but more generally as a system behavior change (change of attractors) as a function of control parameter changes. Since the control parameters we're interested in typically change over time, the difference is subtle and can be glossed over for our purposes. Time can, in fact, be one of the control parameters of the system if attractors are allowed to decay into other attractors. In general, each control parameter is a continuous variable which changes gradually over time, and so scheme trajectories will also tend to be continuous.

"Taboo" areas exist in superdynamical space wherein attractors are not temporally stable. Learning of new knowledge can be thought of as the creation of "well-worn paths" in superdynamical space. Increasingly stringent "taboo-ing" of forbidden regions of superdynamical space to scheme trajectories means that certain behaviors of the dynamical system will eventually never be seen, and that certain combinations of control parameters are never encountered. Conceptually, this means for a given set of control parameters (corresponding to a point on some superdynamical scheme trajectory), the system "expects" the set to change in a way similar to what it has experienced before (the system expects to move along the same superdynamical trajectory as it has in the past). Well-worn paths are those where the surrounding superdynamical energy landscape has become steep. Scheme trajectories will quickly converge to that path.

Upon experiencing a brand new set of environmental inputs, the system will attempt to behave as appropriate to the most similar well-worn trajectory; upon finding that this behavior is inappropriate, a new branch trajectory is found to be appropriate and it too starts to become well-worn. An "experienced" system will have a number of well-worn trajectories, or behaviors, each appropriate to a particular set of environmental inputs.

Inheritance and categorization relations can be seen as varying scales within a basin of attraction

- An inheritance relation between two attractors signifies that one attractor basin is within the other

A common feature of high-level knowledge representations is the inheritance hierarchy in which specific concepts inherit properties from more generalized concepts. In the attractor paradigm, an attractor basin representing a general concept may have smaller basins within it which represent specific concepts. If enough constraints are applied, the system's state will settle in one of the small pits which are highly localized in state space. If the constraints are insufficient to localize the state (resulting in a "fuzzy state"), the system settles into the larger basin. The number and quality of constraints available determine the effective "resolution" of the landscape. An instance of an object in this model is a very local minimum, while a category of objects is a "less local" minimum.

To illustrate, consider the state subspace for a concept such as "car". It will have a certain fuzzy extent along the "length" dimension of 15-25 feet, along the "number of wheels" dimension at four, and so on. If your particular car is a red Chevy, the state subspace is constrained to exactly, say, 23 feet 4 inches in the length dimension and further constrained along the color or red-ness dimension and the "make" dimension. Finally, your car has a license number which constrains it to a unique spot along the "license number" dimension.

This description of the state space provides rich structure at all scales. The model implies that the energy landscape has a somewhat fractal character, in that the roughness of the surface is similar at any scale. Fractal physical structures are common in nature.

Semantic networks can be encoded using a connectionist paradigm

- High-level (conceptual) and low-level (neural) descriptions are isomorphic in linear systems (Smolensky, 1988)

Smolensky contrasts low level descriptions of linear systems with higher level conceptual descriptions. These two levels roughly correspond to the mainstream AI vs. connectionist viewpoints discussed previously. Smolensky distinguishes between distributed representations where a concept is represented by a pattern of activation across all nodes and local representations where an entire concept is represented by a single node (the grandmother cell paradigm). He then shows that the two representations are actually isomorphic in linear systems. The transformation between the two representations corresponds geometrically to a change in coordinate systems. The dynamic evolution of the model is identical no matter which viewpoint is assumed.

Linear systems are a subset of the systems describable using the attractor model. A linear system has an energy landscape which is a hyperplane; the gradient of such a landscape is the same everywhere. A linear system therefore does not have an attractor in the same sense as a nonlinear system. The state of a linear system will, however, settle to the "lowest" corner of a bounding hypercube.

Semantic network encoding of a classic problem in logic

- Dick is a Quaker and a Republican… is Dick a pacifist? (Shastri, 1988)

Shastri proposes a connectionist implementation of a semantic network using the local representation. In this model, nodes correspond to concepts and properties, and connections between nodes correspond to high-level rules or inferences. These rules or connections may be fuzzy, since inference strength corresponds to connection weight. Shastri uses meta-nodes for control of network behavior. For example, binder nodes are used to characterize the inheritance/categorization relations of concept and property nodes. Query nodes are clamped by the user to control what question is being asked of the network.

Operating the network consists of clamping input nodes to input values, and clampling the query nodes to indicate what question is being asked. For example, the input nodes may be clamped to indicate "Dick is a Quaker" and "Dick is a Republican". The query node may be clamped to the question "Is Dick a pacifist?". Encoded in the network connections between concept and property units and binder units are the rules determining probabilities of various inferences. In this example, Dick being a Quaker tends to support the hypothesis that Dick is a pacifist, but Dick being a Republican supports that Dick is not a pacifist. The strengths of the connections as implemented determines the answer that will appear at the output nodes.

The dynamical behavior of the system corresponds to a system with a static, unchanging energy landscape (once input and query nodes are clamped). The network operates on a discrete clock. After a few time ticks, the network settles to an "answer" state, which include the answer node activation representing the correct answer.

Attractors can be used to describe the operation of semantic networks

- Shastri semantic nets do not explicitly address network dynamics, nor does it use a distributed representation

The semantic net connectionist implementation due to Shastri looks at static behavior only. No explicit description of dynamic behavior with changing inputs is addressed. Also, the local representation is not resistant to local damage, requires network topology to be altered to add new concepts, and in general does not display certain properties attributed to biological networks in nature. A distributed representation of concepts, on the other hand, does display these desirable qualities.

- The system does, however, display temporal dynamics with a fixed point attractor for each answer, represented by a final network state

The Shastri semantic net is, however, a dynamical system with a temporal fixed point attractor. When inputs are altered (query or input nodes are changed), the energy landscape changes, the attractor changes, and the system undergoes some dynamic behavior as it settles to the new attractor, eventually again settling to a steady state.

Note that the final (answer) state depends on the initial pattern of activation in the system. The attractors describing system behavior remain the same no matter what the

initial pattern, as long as the inputs are held steady and the network configuration (i.e., connection weights) remain the same.

- The symbolic representation of the Shastri network to solve a given problem does not change much when described using attractors

We can use the attractor metaphor to describe the semantic network by reinterpreting the graph symbols. Using a distributed representation of concepts, each node represents a particular pattern across *all* nodes in the network, rather than denoting a one-to-one correspondence. The connections between these nodes (we'll call them *semantic* nodes as opposed to the physical *network* nodes) then represent attractor transitions rather than a single physical connection. The network becomes a graph representing discrete attractor transitions and stable attractors. The actual physical implementation can use one of the methods described above.

- A semantic node corresponds to the attractor which currently describes the behavior of a subnet

When the network is in a state denoted by one of the semantic nodes, the attractor corresponding to that node describes the dynamical behavior of the subnet – the system will settle to the attractor no matter what pattern of activation was present previously. The main distinction between this description and the local representation description is that a single attractor represents the state trajectory to the answer state in the Shastri net, where the distributed representation implies that the net may transition through several different attractors (and several different dynamic behaviors) before settling into the answer state. In other words, there is a trajectory through superdynamical space as well as through state space.

- Connections between semantic nodes correspond to attractor transitions

An active attractor may transition to the next attractor which may exist entirely in the same subnet, entirely in another subnet, or partially in two overlapping subnets. The semantic connection weight then represents the tendency of a given attractor to change to another. In the attractor model, these semantic weights may themselves exhibit dynamic behavior and asymmetry (as well as network weights).

- What if the network configuration (i.e., connection weight set) changes?

In the attractor model, the network changes configuration as part of normal behavior. When the set of connection weights changes, the energy landscape of the system changes as well as the attractors describing that system. If the initial state of the system (defined by the clamped nodes) remains the same, the final output may change if the initial state lies in a different basin of attraction due to the landscape change. The final attractor in an attractor transition sequence should be temporally stable so that the answer state is also stable.

- Examining the state of the system is difficult

One can at best examine the static pattern of network node activations (a freeze frame) in a subnet and see how similar it is to a known or expected pattern. In the real world, input is continuous and dynamically changing, and may originate from the external

environment or from other regions of the net (other subnets). In this way, any portion of a large network may be extracted for examination, while the remainder of the network is treated as a black box which provides the environment for the subnet being examined. This is a kind of inverted black box view, since everything outside of the subnet being observed is defined only by its interface with the subnet. A subnet is excised from the complete system, its inputs and outputs are defined and controlled or monitored, and the subnet behavior and patterns of activation are examined. The danger of using this method is that the complete distributed representation of a concept may not be included in the subnet.

- The internal behavior of the superdynamical scheme is difficult to interpret directly

The high level superdynamical view is even harder to visualize than the distributed representation of concepts. The energy landscape metaphor may prove useful here, even though it is simplified to only two dimensions at a time. Animated modeling of two-dimensional state and superdynamical subspaces may be helpful for visualization of behavior and its changes over time.

- How might a system based on the attractor paradigm be generated?

We provide here some highly speculative ideas on engineering working systems based on the attractor model. Since the landscape metaphor is useful and intuitive, this provides a good starting point. An editor capable of generating landscapes and their associated attractors in state and superdynamical subspaces would be required.

First, one would define a superdynamical landscape, which relates environmental inputs to the desired behavior of the system given those inputs. The relevant dimensions of this space must be labelled (i.e., determine the relevant environmental inputs). One might define "key frames" of superdynamical scheme trajectories as is done when animating cartoons, to be interpolated by computer. Next, the state space energy landscapes describing the behavior of the system at particular points in superdynamical space would be defined. Much of this task consists of labeling the subspace dimensions with properties. Of course, only a few dimensions are important at any particular superdynamical point. The dependencies between dimensions also need to be defined. One would build a variety of these subspaces to describe the state behavior of the system, to be combined later.

Once the landscapes are complete, they can be combined in the computer into the *n*-dimensional hyperspace. This landscape can then be imprinted onto a fully interconnected recurrent neural network, possibly using a mechanism such as the ones described above. Some connections may be found unnecessary depending on the nature of the task, so they can be eliminated entirely. With this method, connections are not explicitly specified. Rather, they become an internal, distributed mechanism for the network to effect transitions from one state (attractor) to the next based on time or external inputs.

The process of imprinting the energy landscapes onto a physical network needs to be investigated. A particular set of network interconnection weights maps to a single energy landscape, but the inverse mapping may not be unique. There are at least two possible

methods: An analytical transform function may be derived which can generate a set of weights, or the weights may be generated using a learning or imprinting technique over multiple iterations.

Once the connection weights are determined, they could be "burned" into a silicon neural network chip, like burning a PROM. The resulting chip should behave as designed, with far greater speed than a biological neural network, and with the ability to interface directly with conventional computer hardware.

Speculations on problems which may be more tractable using attractor-based descriptions than by semantic networks

This section discusses several problems which traditional artificial computational systems have had difficulty solving. Attractors may help provide elegant solutions; we venture guesses here on how certain properties of attractors might help in solving some of these problems and in designing truly general-purpose computers.

- **Handling the novel, unexpected input which can occur in open systems**

In a Shastri network, unexpected input requires creating and adding new conceptual nodes and meta-nodes and encoding new categorization/inheritance relations. This can be a complex ad hoc task even when automated. However, in an open system, new concepts should be add-able at any time without major restructuring of the network. Since attractors do not have to be explicitly described to exist, new attractors may be formed and incorporated by the system spontaneously simply by modifying a particular locale in the energy landscape, while leaving the rest intact.

The attractor network can accept input from another subnet or from the external environment in exactly the same manner. No distinction is made as to the source of input. Excitation of a pattern in a subnet can arise from external stimuli or from excitation in a nearby or overlapping subnet. There are no explicit bounds to the attractor description (subnets may be any size), making it suitable for a hierarchical, scalable open system description. Networks can be expanded to include any size organization while still using a similar description. Sociological studies have indicated that many tasks in large organizations are repetitive in nature – perhaps even these high-level behaviors may be describable using periodic attractors.

- **Periodic behavior modified by changes in external input**
- Walking or running is a behavior in nature which may be describable by attractors

The attractor method generalized to periodic attractors suggests several potential applications. Walking or running can be implemented by a cyclic activation of sets of motor neurons (Arbib, 1985). Different gaits are slight modifications of the generalized periodic walk/run behavior. One can imagine a walk/run attractor modulated by a "desired speed" control parameter. Once the control parameter is set, system behavior settles to the walk/run attractor causing that cycle to repeat until the control parameter changes. At a certain speed, the periodic walk attractor transitions to a slightly different run attractor.

- Music is a perceptual behavior which seems to create some kind of precisely timed repetitive cycle

Music is another set of behaviors which suggest the presence of a periodic "rhythm attractor". This attractor might be generated through synchronous reinforcement with auditory sensors. A trained musician would be able to further translate this sensory attractor into a motor attractor and thereby generate precisely timed, metronome-like movements of the fingers.

If this viewpoint is examined closely, some interesting aspects are noted. The music attractor needs to display minimal jitter (i.e., it must begin its cycles at precise times; the interval does not vary between cycles), which would demonstrate the ability of the brain to maintain precise timing. The modulation of the repetitive patterns of both time (to form rhythm) and pitch (to form scales and harmony) in a myriad of different ways is the basis of music.

There seems to be something very fundamental about music and its connection to natural computation in the human brain. Both are intellectual activities which seem simple to those who engage in them, but both become very complex upon analysis. Perhaps the well known structure of music can provide clues to the little known hidden structure of information processing in the brain.

- Periodic attractors are a natural description of repetitive behaviors

There are many types of cyclic behavior seen in life, from the simplest form of motor movement in simple creatures to the repetitive daily cycles of large social organizations. The actual periodic behavior seems to become automatic after learning; this repetitive, "mindless" behavior can then be modulated to fit the particular circumstances of the environment.

- Chaotic attractors could describe certain random aspects of natural computation

What role might chaotic attractors play in computation? Perhaps the brain is extremely capable of modulating attractors (i.e., moving about in superdynamical space), including transitioning deliberately from fixed point to periodic to chaotic attractors. When a particular problem solving approach does not work, could it be that chaos is introduced to provide innovative alternatives? Those questions and many more which the attractor model brings up cannot be answered without a great deal more research.

Conclusion

The attractor model of natural computation seems well suited for describing highly complex dynamical systems such as neural networks. This field has been researched very little to date. It is highly interdisciplinary, taking bits and pieces from computer science, mathematics, physics, biology, psychology, and sociology. The model presented here is far from complete. For example, little attention has been given to the method of transforming between the various representations presented. However, the new perspective provided by this model and the description of potential interpretations are a first step in that direction. There are innumerable opportunities for further research along these lines.

Bibliography

Abraham, Ralph & Shaw, Christopher (1988). Dynamics – The Geometry of Behavior, Part 4: Bifurcation Behavior. *Aerial Press.*

Amit, D. J., Gutfreund, H., & Sompolinsky, H. (1985). Spin-glass models of neural networks. *Physical Review*, 1007.

Arbib, M. A. (1985). Schemas for the temporal organization of behavior. *Human Neurobiology, 4*(2), 63-72.

Bell, T. (1989). Sequential processing using attractor transitions. *Proceedings of the 1988 Connectionist Models Summer School* (pp. 93-102). Morgan Kaufmann.

Buhmann, J., & Schulten, K. (1988). Storing sequences of biased patterns in neural networks with stochastic dynamics. In *Neural computers* (pp. 231-242). Berlin: Springer Berlin Heidelberg.

Kleinfeld, D., & Sompolinsky, H. (1988). Associative neural network model for the generation of temporal patterns. Theory and application to central pattern generators. *Biophysical Journal 54 6, 54*(6), 1039-51.

Minsky, Marvin (1985). The Society of Mind. *Simon and Schuster.*

Peretto, P., & Niez, J.-J. (1986). Stochastic Dynamics of Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics, 16*(1), 73-83.

Schreter, Z. (1988). Sequential processing by overlap and fatigue of memories. *Neural Networks, 1*, 218.

Shastri, Lokendra (1987). A connectionist encoding of semantic networks. *Distributed Artificial Intelligence, 177-202.*

Shastri, L. (1988). A connectionist approach to knowledge representation and limited inference. (Elsevier, Ed.) *Cognitive Science, 12*(3), 331-392.

Smolensky, P. (1986). Neural and conceptual interpretation of PDP models. *Parallel Distributed Processing.* 390-431.

Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and brain sciences, 11*, 1-74.