A Logical Topology of Neural Networks

Alianna J. Maren
Accurate Automation Corporation
7001 Shallowford Road
Chattanooga, TN 37421

Proc. Second Workshop on Neural Networks:
Academic/Industrial/NASA/Defense (WNN-AIND91)
Auburn University

February 11-13, 1991

Abstract

The field of neural networks has evolved sufficient richness within the last several years to warrant creation of a "logical topology" of neural networks; a conceptual layout of the relationships between different neural networks. By identifying the fully-connected network of continuously-valued neurons as a logical "North Pole," and using the increase in network structure and specificity as a qualitative "distance" between a network and this pole, we achieve a preliminary topology. Using structural similarity as a basis for making distinctions, we obtain five distinct categories of neural networks. Within a category, networks are not only structurally similar, but also have similarities in dynamics, learning rules, and applications.

Interesting networks with potentially novel properties can be created by combining basic aspects of different network classes. Examples are the Boltzmann machine, which combines the energy-function minimization of a laterally-connected (Hopfield-type) network with the structural organization of the multilayer feedforward (Perceptron-type) networks, and the masking field network, which embeds a cooperative-competitive second layer into a classic ART 1 architecture.

1 A Rationale for a Logical Topology

As any discipline matures, one of the steps its practitioners undertake is to form a topology for that discipline; to lay out logical relationships between its various components, and to identify what structures emerge from the initial seeming chaos of the discipline's early years. The task of developing a coherent internal structure is crucial to the establishment of a discipline. An example of developing this structure can be found in the history of the Artificial Intelligence (AI) community.

About ten years ago, in the heydey of artificial intelligence, there were strong opinions regarding the relative role and importance of different aspects of AI. Daniel Dennett [1] relates the following story:

> "... in a heated discussion at MIT about rival theories of language comprehension, (Jerry) Fodor characterized the views of a well-known theoretician as 'West Coast' – a weighty indictment in the corridors of MIT. When reminded that this maligned theoretician resided in Pennsylvania, Fodor was undaunted. He was equally ready, it turned out, to brand people at Brandeis or Sussex as West Coast. He explained that just as when you're at the North Pole, moving away from the Pole in any direction is moving south, so moving away from MIT in any direction is moving West."

Dennett then used this story to organize a somewhat tongue-in-cheek "logical topology" of artificial intelligence, with MIT, the "Vatican of High Church Computationalism," as the North Pole, and with the different aspects of West Coast "Zen Holism" distributed along the 360° horizon. Figure 1 reproduces Dennett's world-view of AI at that time. The neural network community is well-represented with such figures as Fahlman, Hinton, Smolensky, and McClelland; even Feldman can be considered a connectionist. It is interesting to note, though, that Dennett's organization places the neural networks community only 90° away from Marvin Minsky, whose early comments had such an influence on the field.

At the time when Dennett wrote is first draft of the article in 1984, members of the neural networks community were still out beyond the pale; part of the West Coast "New Connectionism." But since the 1986/87 re-emergence of neural networks as an identifiable discipline, we have taken on a size and diversity equivalent to that of the AI community ten years ago. In this light, it may be to our advantage to examine the landscape – the *topology* – of the different neural network approaches and paradigms.

The Logical Geography
of Computational Approaches
(A View from the East Pole)

Daniel C. Dennett

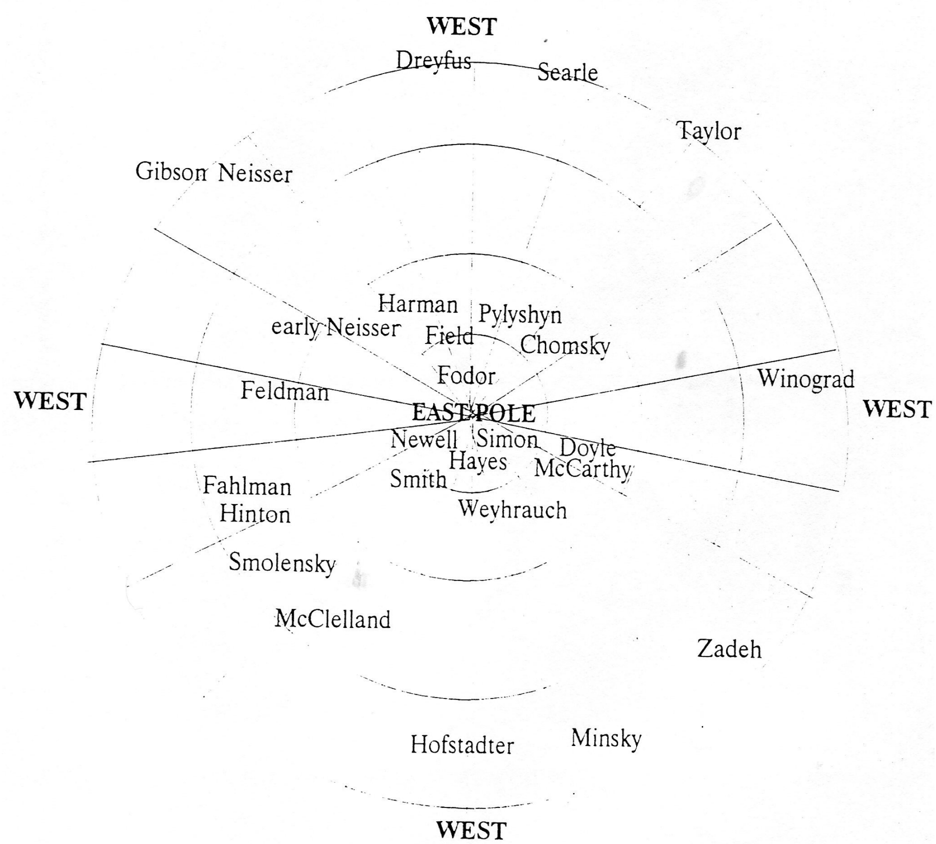

Figure 1: Logical topology of computational approaches, by Daniel C. Dennett. (Permission granted to reproduce.)

By creating a topology of neural networks, we obtain several advantages. First, we have a greater sense for the relationship between distinct neural networks. "Closeness" in a topological sense should indicate some similarity not only between network structures, but between dynamics and learning methods as well, since both dynamics and learning are intimately related to structure. This "logical topology" approach allows us to more clearly identify properties of a class of networks (networks which are clustered together in a topological relationship). The existence of class properties suggests that subtly different networks within a class can often be used for the same task or application, but with different performance characterizations. Comparative evaluations of networks within a class will be useful in ascertaining these differences. Finally, such a topological organization suggests the basis for both a more formal description of networks and their properties.

Within the past several years, there have been various efforts to build some classification or ordering of existing neural networks. None so far have produced a useful and adopted taxonomy, although the various distinctive neural network paradigms are by now largely understood by most members of the neural networks research community. Thus, some effort to create a "logical topology" for neural networks is now in order.

2 What is North?

If we accept the idea that a logical topology of neural networks is a reasonable thing to develop, then our first question is equivalent to: *What is North?* Dennett humorously organized the main computational approaches of AI around a mythical "East Pole" centered at MIT. Can we even envision a similar central point for establishing a topology of neural network systems? Further, can we identify some concepts of directionality and distance between neural network paradigms? To do so, it helps to look over the (recent) history of neural networks and identify the efforts which others have made. Specifically, we search for and evaluate the distinctions which others have used in building a classification of different neural network systems.

In 1984, when neural networks made their most recent re-emergence as a computational discipline, there were only a few types of neural networks which were commonly known. As an example, four different types of networks were described in the now-classic *Parallel Distributed Processing* [2]. The next year, Richard Lippmann's "An introduction to computing with neural networks" [3] was an underground hit at the first *International Conference on Neural Networks*. At this time, Lippmann's paper was the only one to review a variety of different network types.

Lippmann made two primary distinctions: (1) whether a network had supervised or unsupervised learning, and (2) whether it operated on binary or continuously-valued data. We now know many networks that, with suitable modifications, can operate on both binary and continuous data (e.g., the discrete and adaptive Hopfield-Tank networks, and the two main versions of Adaptive Resonance Theory networks; ART 1 and ART 2). Thus, making a network-class distinction based on input data type is not very useful. Further, some networks (e.g., the Learning Vector Quantization network and the Neocognitron) can learn in either supervised or unsupervised modes. Thus, supervised vs. unsupervised learning is not a good primary distinction either.

Shortly after Lippmann published his work, Robert Hecht-Nielson published a review article describing thirteen different types of neural networks [4]. Although he listed the unique characteristics of each of these thirteen networks, he made no effort to form a taxonomy.

The following year, Patrick Simpson began circulating a pair of comprehensive review papers which described a total of 26 neural networks [5]. (This pair of papers was published as a book in 1990 [6]). Simpson's review papers (1988), and later his book (1990), classified networks into one of four groups based on two distinctions espoused earlier by Bart Kosko [7]. One distinction was between networks with supervised vs. unsupervised learning, and the other was between feedforward and feedback networks. Again, this classification scheme is not useful across the broad range of networks. As just discussed, the distinction between supervised and unsupervised learning is not a useful primary distinction between network types. The distinction which Kosko made between feedforward and feedback networks is not generally useful; it highlights his Bidirectional Associative Memory network without giving appropriate consideration to the many networks which do not fit well within such a simple structural classification schema (e.g., the Hopfield-Tank network [8]).

In reviewing the neural networks textbooks now available, and the literature accumulated over just the past five years, it is easy to identify about four dozen distinct neural networks. Many of these are described in the *Handbook of Neural Computing Applications* [9], including about a dozen networks which are not described in the Simpson book. This is clearly a case of exponential growth, partially illustrated in Figure 2. However in most of the summary literature and the many introductory textbooks which exist now, the different neural networks are presented in a "stand-alone" manner, with few efforts to create useful conceptual classes of networks.

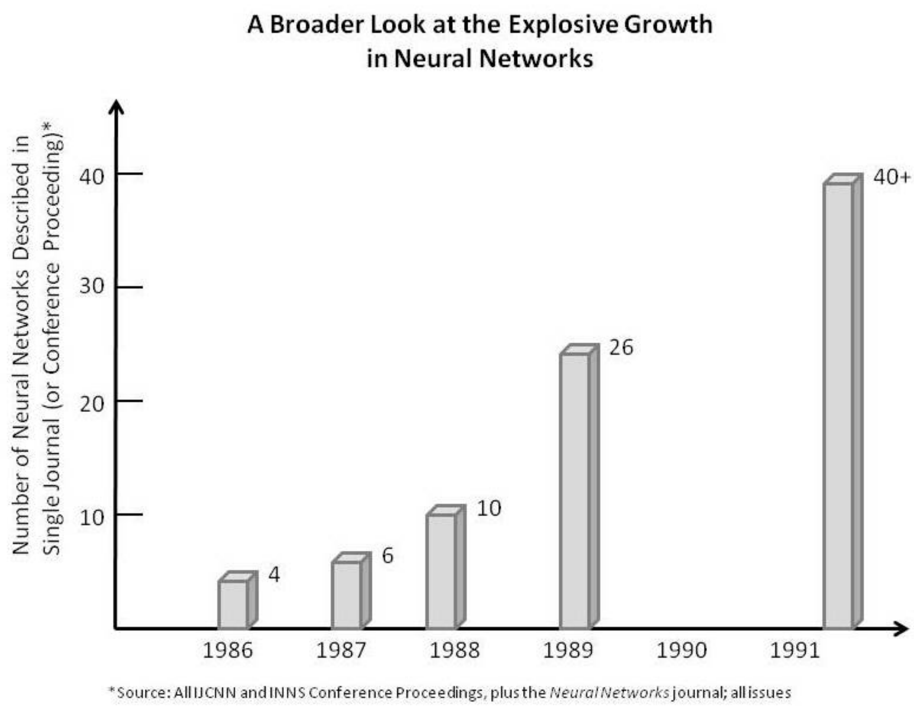

Figure 2: Explosive growth of neural networks: 1986 - 1991. Data collected by author from sets of *IJCNN* and *INNS Conference Proceedings*, as well as the *Neural Networks* journal.

In creating a logical topology of neural networks, it is useful to make a distinction between different levels of description of a neural system. In the *Handbook* [9], I introduced three levels useful in describing neural networks:

- ***The micro-structural level***, for describing the composition of an individual neuron or other component of the neural network,
- ***The meso-structural level***, for describing a neural network itself in terms of observables such as numbers of layers, connectivity patterns, etc., and
- ***The macro-structural level***, for describing systems of interacting networks.

The logical topology presented in this work is at the meso-structural level, and in fact is specifically concerned with laying out a topology in terms of network structure. It is worth noting that the topology proposed here thus does not extend to macro-level descriptions, as would be appropriate for describing such systems as Lapedes and Farber's Master/Slave network [10], Werbos' Adaptive Critic [11, 12], or Carpenter and Grossberg's systems of ARTs (ART3) [13].

As we examine an approach to conceptual network topology based on network structure, we find that network dynamics are intimately related to structure, and that certain types of learning are often associated with certain structures and their dynamics. Further, we find that network functionality (what it does) follows readily from its structure, which reminds us of the old axiom, *form follows function*. This selection of network structure serves as the primary basis for making distinctions among networks and for creating classes of networks. It also gives us some basis for establishing a qualitative measure of "distance" between two different neural networks. However, with the selection of structure as the basic element of our topology, the question remains, *What is North?*

3 The Fully-Connected Recurrent Network: A "Logical" North Pole

In order to identify a logical "North Pole," one that would help in creating a topology, it is useful to recall the developmental history of neural networks. This is because even though we are now able to create a set of logical relationships between different neural networks, there is a strong sense of history to the development of different neural networks. In fact, during the neural network "glacial period" initiated by Minsky and Papert's publication of their book *Perceptrons* in 1969 [14], many fine scientists developed neural network approaches in relative isolation. This led to distinctly different "phylogenetic branches" of neural network evolution. Together, Figures 3 and 4 illustrate the major lines of development.

The very earliest neural networks were very small and highly structured, as described by McCulloch and Pitts [15]. Rosenblatt generalized the layered structure implicit in the early McCulloch and Pitts sketches to form the Perceptron [16]. Widrow and Hoff used an identical architecture in forming

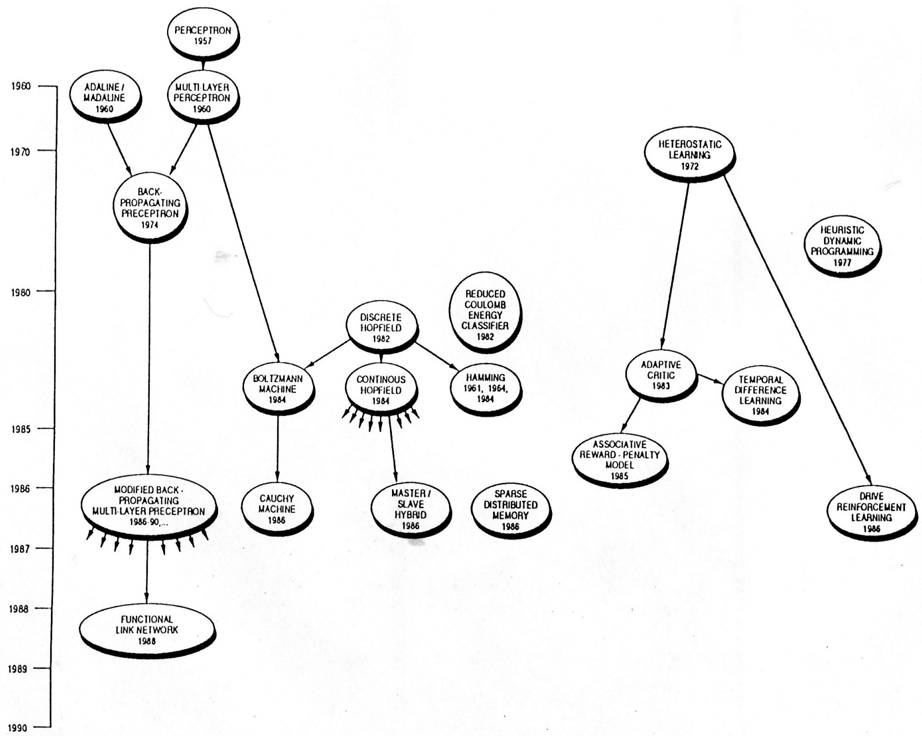

Figure 3: Historical development of well-known neural networks: I

the ADALINE/MADALINE networks [17]. These networks formed the basis for evolving the phylogenetic branch of multilayered feedforward neural networks. These networks were characterized, especially in the early years, by strict feedforward connections. (We note, though, that the concept of recurrent connectivity was advanced by McCulloch and Pitts, and is not a recent introduction.)

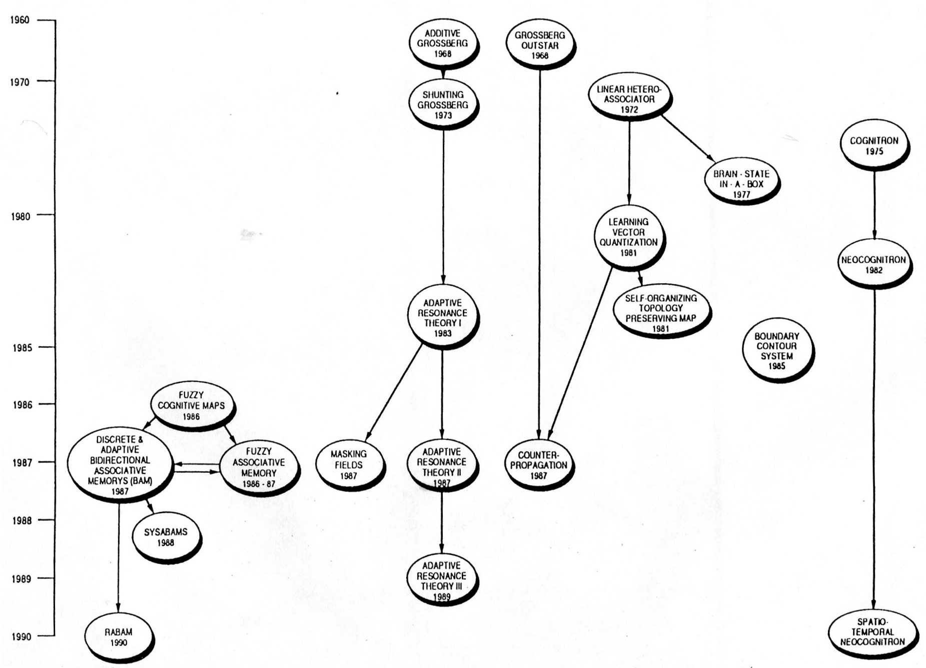

Figure 4: Historical development of well-known neural networks: II

In contrast to the development of layered feedforward networks, an entirely different line of inquiry was taken by researchers (mostly physicists) who investigated the properties of systems of fully-connected McCulloch-Pitts neurons. (See [18] for review and references.) This led to elaboration of the basic McCulloch-Pitts neuron into a "noisy" neuron, which was investigated by Little [19] and others. A network composed of these noisy neurons is *ergodic*, or one which can inhabit, over time, all possible states. Hopfield [20, 21] investigated the dynamics of a system of hard-threshold, non-noisy neurons and demonstrated non-ergodic memory-retrieval properties.

A third major line of inquiry (alongside the multilayer feedforward networks and the fully-connected networks) was established by the independent work of Anderson [22] and Kohonen [23] in studying the associative properties of networks of linear neurons (as distinct from the McCulloch-Pitts bistate neurons). A fourth branch in the evolution of major systems of neural networks was initiated by Grossberg, Carpenter and colleagues [24, 25], leading to the system of Adaptive Resonance Theory (ART) networks from the 1970's to the mid-1980's [26, 27]. Thus, by the early 1970's, there were at least four major evolutionary branches of neural networks.

On examining these different evolutions of neural network methods for a suitable "North Pole," we find that the fully-connected network of analog neurons is the most general neural network system. All other neural networks can be created by introducing constraints on this structure, which may be at the neuronal (micro-structural) level, or at the network (meso-structural) level. An example of a constraint at the neuronal level is to move from the neuron with continuous output values to the bistate neuron. An example of a constraint at the network level is to partition the network into two fields, and allow connections only between the neurons in the different fields (e.g., the basic architecture for a linear associator).

Having established the fully-connected network of analog neurons as the logical "North Pole" of a neural network topology, it is reasonable to think of "distance" from this "pole" in terms of the degree of structuring (decomposing into layers, restrictions on connectivity, etc.) introduced into a specific network. To make this specific with examples, both the Brain-State-in-a-Box and the Hopfield-Tank network are very close to the North Pole. The Perceptron is further away, the Adaptive Resonance Theory networks and the Kohonen networks are much further "south," and an exotically-structured neural network such as Edelman's Darwin III is off the map. Figure 5 illustrates a preliminary sketch of the topological relations between networks. Although neighbor relations (expressed as connecting lines) are valid, the distance measure between networks is variable.

4 The Five Major Neural Network Classes

We find that, simply on a structural basis, we can identify five different classes of neural networks, as depicted in Figure 6:

- ***The multilayer feedforward networks***, exemplified by the Perceptron architecture,

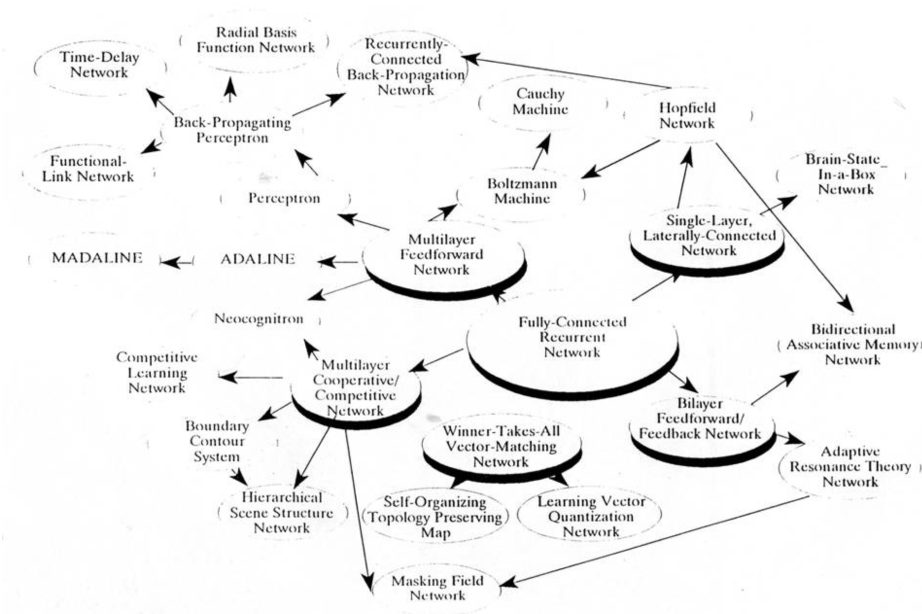

Figure 5: Topology of neural networks

- ***The single-layer laterally-connected networks***, exemplified by the Hopfield (or Hopfield-Tank) network,
- ***The bilayer feedforward/feedbackward networks***, exemplified by the ART networks,
- ***The topologically-organized vector-matching networks***, exemplified by the Kohonen networks, and
- ***The cooperative-competitive networks***, exemplified by Grossberg et al.'s Boundary Contour System (BCS).

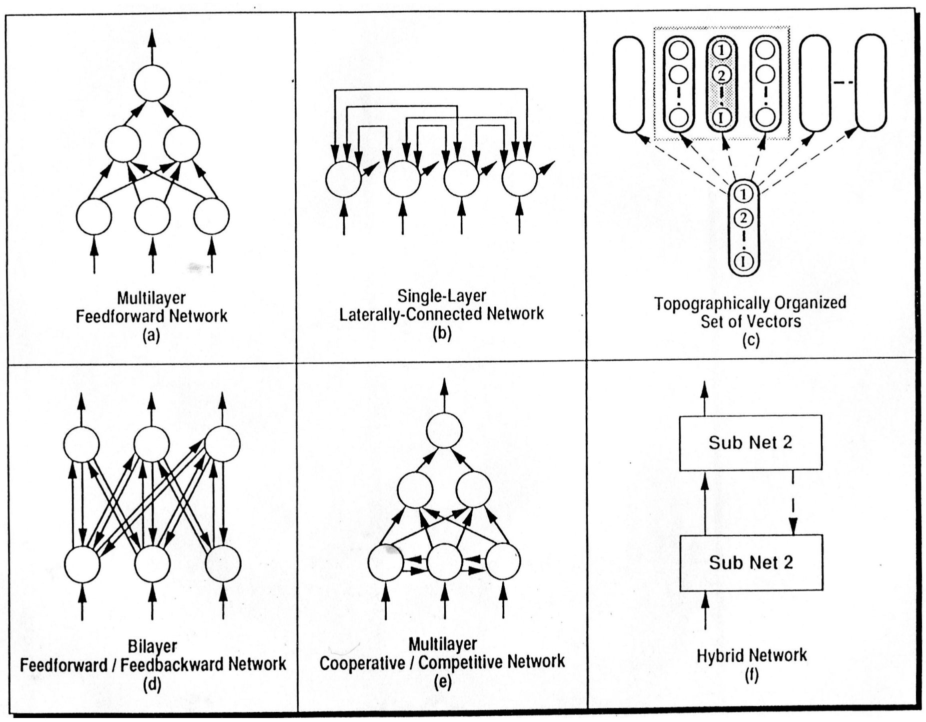

Figure 6: Six mesostructures for neural networks

The first three of the classes itemized above are well-known, and the last two are less well-known, although the Kohonen networks are rapidly gaining more prominence.

In addition to these five major categories, there is a final category for hybrid networks which cannot be readily classified into one of the five major categories, because they incorporate a blend of two or more primary structures.

An example of this is the Hamming network [28], which introduces a Hopfield network into the output layer of a Perceptron in order to drive the output towards a known state. With the exception of the cooperative/competitive and the hybrid classes, which are recent in their evolution, all other neural networks classes have a development history of approximately 25 years each. These basic neural network class typologies are illustrated iconically in Figure 6.

We note that networks with similar structures often have similar dynamics, similar learning rules, and can handle similar tasks. For example, all multilayer feedforward networks that do not have cooperative-competitive lateral interconnections undergo supervised learning, and can all perform pattern recognition and classification tasks.

4.1 Multilayer, Feedforward Networks

The class of multilyaer feedforward networks includes the Perceptron [16], its close relative the ADALINE/MADALINE, and more structurally-complex networks such as the radial basis function network [29], the functional link network [30], sigma-pi networks [2], and various time-delay networks. The most common learning method is backpropagation, developed by Werbos in 1974 [31] and later presented (although without attribution to the initial developer) by Rumelhart et al. in 1986 [32]. These networks comprise the most well-known and commonly-used class of neural networks for both research and applications. Because knowledge of these networks is so common, no further discussion of this class is necessary.

4.2 Single-Layer, Laterally-Connected Networks

The most well-known exemplar of this network class is the Hopfield (or Hopfield-Tank) network [8, 20, 21]. It is less well-known that Anderson's Brain-State-in-a-Box (BSB) network is almost structurally identical to the Hopfield network [33]. As single layer, laterally-connected networks, these are also functionally similar, in that both perform autoassociation. However, the Hopfield network can also perform optimization tasks [8].

Although similar, the Hopfield and BSB networks differ at both the micro-structural and meso-structural levels. At the micro-structural level, the discrete Hopfield network has bi-state neurons, the continuous Hopfield network has neurons with continuously-valued output in the range (0,1), and the BSB network has linear neurons with lower and upper activation hard-limits fixed at -1 and 1. At the meso-level, each neuron in each type of network is fully connected to each of its neighbors, but the BSB neurons are recurrently connected as well. However, the main difference between these two networks is in their dynamics; the BSB network eases neurons away from saddle points towards learned energy minima, so avoiding some of the "spurious minima" which so plague the Hopfield network [34].

The Hopfield network, unlike the BSB network, can also be used to perform optimization tasks. This is because an alternative view of the Hopfield network dynamics is that it operates to minimize an energy function, and the only requirement of the energy function is that it meets Lyapunov criteria [20, 21]. This allows the energy function to be constructed so as to reflect the constraints of many interesting optimization tasks [8].

One of the most interesting distinctions between networks of this class and the multilayer, feedforward networks is the difference in the stopping criteria for network learning and/or dynamics. The stopping criteria for processing in a feedforward network is that the flow of activations from one layer to the next simply reaches the last layer. Since there is no place for further movement, the dynamics perforce stop. In contrast, in a single-layer, laterally-connected network, there is no *structurally-embedded* stopping criterion. The stopping criterion is imposed externally, by having some daemon observe that continued repetition of the dynamic processes no longer induces change in the network's state.

In most cases, it is desirable that the network does converge to some stable state. Because this is a desired property, much work on the dynamic properties of a single-layer, laterally-connected network proves (or occasionally disproves) the *convergence properties of the system dynamics*. This is necessary since the dynamic properties are not intrinsically bound by system structure, and must be externally defined. This quality pertains also to the next neural network class; the bilayer recurrently-connected networks. In these networks also, the dynamics are constrained by properties other than network structure, and dynamic convergence is again an important issue.

4.3 Bilayer, Recurrently-Connected Networks

This class of neural network, typified by Carpenter and Grossberg's Adaptive Resonance Theory (ART) networks [26, 27] and by Kosko's Bidirectional Associative Memory (BAM) networks [7], are a logical outgrowth of early work on the linear associator, independently developed by Anderson [22] and Kohonen [23]. All of these networks use a structure consisting of two layers; an input layer and an output layer. The output layer associates patterns or activates a classification node in response to different input patterns. The linear associator had a single connection weight matrix, and both the ART and BAM networks use two connection weight matrices; one forward and one backward.

The most interesting aspect of the bilayer, recurrently-connected networks is that they operate with a continuous dynamic, much like the Hopfield network. In fact, the BAM network brings together the structural design of a bilayer network (e.g. the linear associator, but adding in a feedback layer as in Carpenter and Grossberg's ART network) with the dynamics of a Hopfield network. The limitations of this approach are that the BAM is subject to spurious recall and has limited memory capacity, much like the Hopfield network [34]. Because the dynamics require that the neural network converge to some state in order for processing to stop, much attention in this class of networks is given to stability considerations.

4.4 Topologically-Organized, Vector-Matching Networks

The class of topologically-organized, vector-matching networks is perhaps the most phylogenetically distinctive neural networks class, initiated by Teuvo Kohonen [35, 36]. These networks are also the most difficult for someone who works with "typical" neural network processes to understand. This is because the operations are all performed strictly on vectors that represent the "connection weights" of a neural network, but with the actual neurons (to which these weights would normally connect) removed from the system. The vector operations include use of a dot product to measure similarity between an input vector and each of the existing network vectors (set of "connection weights"), and further operations that modify the vector values of those vectors that are within a certain distance of the "winning vector;" the one that is found to have the closest vector distance to the input. Two well-known exemplars of this class are the Learning Vector Quantization network and the Self-Organizing Topology Preserving Map. The former has been used successfully for many clustering applications, and the latter extends the concept of the former into a smoother mapping of different inputs.

4.5 Multilayer, Cooperative/Competitive Networks

The cooperative/competitive networks are distinctive from their topological nearest-neighbors; the feedforward networks. However, the distinction lies more in the concept provoking the network design rather than in the explicit structural organization of the networks. The reason is that the cooperative and competitive effects are formally designed as connections between neurons at the same level in a network structure. They could alternatively – with a different configuration – be implemented in a feedforward manner. This alternative configuration builds on the recognition that any arrangement of lateral connections may be redesigned in a feedforward sense, where the individual nodes are replicated at the next higher level, and the connections that originally were lateral ones are now redesigned as structural feedforward connections. This is a fine distinction, but has served to create a number of networks with unique and very interesting properties.

Grossberg and Mingolla's Boundary Contour System (BCS) network [37, 38], developed for image segmentation, is perhaps the most well-known exemplar of this class of neural network. Fukushima's Neocognitron can also be considered to be a cooperative/competitive network, due to the insertion of subsidiary inhibitory cells in the feature extraction layer [39]. An early cooperative/competitive network, which had limited capabilities, is the Rumelhart-Zipser competitive learning network [40]. Maren et al. [41, 42] have developed a cooperative/competitive network which is useful for assignment tasks.

In particular, Maren and Minsky [43] have developed a cooperative/ competitive means of creating a perceptual structure for images using the image segments as building blocks. This approach is somewhat akin to Grossberg and Mingolla's work, but operates at the perceptual grouping rather than the low-level boundary contour level. It also builds a full hierarchical structure for the image, rather than simply completing low-level units. The resulting hierarchical structure can be decomposed by a separate process to yield hierarchical sub-structures that can be ascribed to man-made or natural objects, taking into account perceptual grouping principles.

5 Bridging the Gap: Networks which Derive from Neighboring Classes

Some of the most interesting networks are those which have been developed out of concepts underlying different classes. Examples of these are the Boltzmann machine (and its derivative, the Cauchy machine) and the Masking Field network. This section describes these as illustrations of how cross-class networks can emerge and possess interesting hybrid properties.

5.1 The Boltzmann Machine

The Boltzmann machine [44, 45] is an example of a network which bridges two classes in an interesting and effective manner. That is, the Boltzmann machine offers interesting properties which are not available to either of its predecessor networks, and its performance is acceptable.

The Boltzmann machine and the back-propagating Perceptron, both of which became popular at about the same time, have almost identical structures. (The differ only in that the Boltzmann machine neurons are hard-limit, whereas the output of a back-propagating Perceptron's neurons are continuous.) With the exception of the difference introduced by the two different neural transfer functions, the dynamics of each network's operations are also identical. Yet there is a key difference between these two networks which manifests in terms of their learning rule, and which has its antecedents in the developmental history of each network.

Whereas the back-propagating Perceptron is a direct linear descendant of the Perceptron, the Boltzmann machine actually grew out of the Hopfield network architecture. The basic, single-layer network was conceptually separated into two fields (input and output), much as was done later in the BAM. The difference is that "hidden" neurons were introduced into the Boltzmann machine; these connected the input and output layers. As a result of the evolution from the concept of a fully-connected network, the Boltzmann machine was created with a learning rule much more akin to a modification of a Hopfield network learning rule than to the back-propagation network. (The simulated annealing used within the Boltzmann machine learning is simply a way to obtain energy function minimization.) Thus, the Boltzmann machine (and its descendant, the Cauchy machine [46]) can perform optimization - a function usually associated with the Hopfield network [47, 8].

The Boltzmann machine and the Cauchy machine are the only feedforward neural networks which can perform optimization tasks [9, 18, 45, 46, 47], as well as the classification/mapping tasks usually associated with feedforward networks. This capability comes about through combining the learning method associated with the single-layer, fully-connected networks with a more layered structure.

5.2 The Masking Field Network

The masking field network, developed by Cohen and Grossberg [48], combined with the basic ART network, is offered as further illustration of how combining characteristics of different network classes can yield a new hybrid network with interesting properties. In this case, the new property which is afforded is temporal persistence of memory, or recoding a temporal pattern sequence as a spatial pattern of different activation strengths. In this, it is somewhat akin to Edelman's Selective Recognition Automata [49].

The masking field network combines cooperative-competitive structures and dynamics with the basic bilayer structure. The first layer is the input layer (which may incorporate the input, buffer, and feedback storage components of an ART first layer). The connections between the input and the second layer (the masking field) are feedforward. (When the full ART architecture is invoked, feedback connections are used as well.) The masking field nodes each contain a recurrent self-connection and a set of inhibitory connections to all other nodes in that layer. The self-connection embodies cooperative stimulus, whereas the lateral inhibitory connections comprise the competitive aspect.

The recurrent connectivity within the masking field allows temporal persistence of activation, something which is observed in few networks. As different second layer nodes respond with different degrees of (slowly decaying) activation, a pattern of activity emerges across the masking field layer which corresponds to the temporal order of pattern presentation in the input layer. This led Cohen and Grossberg to remark on the predictive aspects of the masking field architecture. It may be more significant to note that this architecture lends itself well (as do varieties of the Selective Recognition Automata [49]) to recoding of temporal sequences as spatial patterns. There are very few networks which have this capability. By embedding a masking field second layer in an ART system, Cohen and Grossberg obtain the self-scaling attributes of an ART classifier, with the additional benefits of temporal pattern sensitivity. This is an excellent example of building a new neural network with elements extracted from the different major network topologies.

5.3 The Cortical Engine

One of the most interesting challenges which emerges from this depiction of the "logical topology" of neural networks arises from examining the distinction between the feedforward networks and the laterally-connected networks. The final state of the feedforward networks is determined by a structurally-embedded dynamic, and the final state of the laterally-connected networks is determined by minimizing an "energy function." This led us to ask about the possibility of creating a neural network that had pattern responsiveness (in the manner of feedforward networks) but which also operated under a continuous dynamic, as with the laterally-connected networks. The advantage of the latter is that it provides a means for continual operation of the network, even when an input pattern is not present. This type of process has been used to create networks which operate continuously, through time, as in the case of some central pattern generators [50]. When an input pattern is presented to such a neural network, it acts to "reset" the continuously-evolving process which manifests as pattern activations in the network.

A ***new class of neural network*** is being developed, the "cortical engine," which incorporates feedforward pattern recognition as mediated by a continuous minimization of a free energy in the "computational layer" of the network. This work builds on a foundation laid by Cowan [51], who suggests that statistical mechanics processes, including free energy minimization, very likely underlie neural activity. This connection to actual neural systems is a step beyond the model invoked in the Boltzmann network, which does not identify any association with real neural systems.

This *cortical engine* network, along with the Hamming network, exemplifies a type of hybrid in which the input connections are feedforward and the activations in the pattern recognition layer are modified by a dynamic process. The cortical engine presents a significant advance in that it ***exhibits continuous processing, whether or not an input pattern is being presented to the network***. Factors which keep the network from "freezing" into a free energy minimum include introduction of network noise along with activation decay. The free energy term includes an entropy term which governs not so much the distribution of nodes among the "on/off" states, but rather the distribution of "on" and "off" nodes into pattern clusters using the ***Cluster Variation Method*** [52]. This drives the network towards diversified patterns as stable states. This network, when suitable lateral interactions are introduced, becomes capable of interesting temporal association behaviors.

6 Summary

A "logical topology" of neural networks is centered around a conceptual "North Pole;" a fully-connected network of continuously-valued neurons. By imposing different structural constraints on this very general network, we obtain the basic forms of five different classes of networks:

- The multilayer feedforward networks,
- The single-layer laterally-connected networks,
- The bilayer recurrently-connected networks,
- The topologically-organized vector-matching networks, and
- The cooperative-competitive networks.

Within each class of network, we find similarities in learning, dynamics, and applications capabilities. These network classes, along with their most well-known exemplars and primary applications uses, are summarized in Figure 7.

The topology presented here is limited to descriptions of neural networks at the meso-structural level, and thus addresses neither networks which are created via introducing complexities into either the neuronal model or the connection weights, nor the many interesting systems of neural networks which have taken on greater prominence in recent years. Further, this particular topology does not differentiate among network types which involve different orders of neural connectivity. These areas are possible extensions of this topology.

Despite these limitations, this topology does suggest some fruitful areas for research. Often, the greatest advances are made, not by working within a single discipline or modality, but by cross-fertilization. The preceding section illustrated three cases which demonstrate the value of this approach, as applied to building networks which incorporate aspects of two different major

STRUCTURE:	Single Layer, Laterally-Connected	Topological Arrangement of Vectors	Bilayer, Feedforward/ Feedback	Multilayer, Feedforward	Multilayer, Feedforward &/or Feedback &/or Laterally-Connected
DYNAMICS:	Recurrent	Vector Matching	Recurrent	Feedforward	Cooperative / Competitive
NETWORK TYPE:	Hopfield Brain-State-in-a-Box	Adaptive Vector Quantization Learning Vector Quantization Self-Organizing Topology-Preserving Map	Bidirectional Associative Memory Adaptive Resonance Theory	Basic Perceptron ADALINE / MADALINE Back-Propagating Perceptron Boltzmann Machine Functional-Link Net	Competitive Learning Net Masking Field Neocognitron Boundary Contour System Hierarchical Scene Structure
APPLICATIONS USES:	Autoassociation Optimization (Hopfield)	Autoassociation Data Compression Optimization (LVQ)	Heteroassociation Pattern Recognition	Heteroassociation Pattern Recognition (spatial, temporal, & spatio-temporal) Data Compression Signal Filtering Optimization (Boltzmann Machine) Image Processing	Pattern Recognition (spatial, temporal, & spatio-temporal) Image Processing

Figure 7: Structure and applications of neural networks

network classes. By extension, the confinement of this topology to the meso-structural descriptive level points out the possibility of attending to neural networks which exhibit cross-scale interaction. An example of this would be to create neural elements with greater internal complexity, and to make the response characteristics of these neurons tunable in response to events at the neural network level. Such cross-scale interactions introduce many interesting issues, such as the mechanisms for interaction, relative temporal scales for dynamics, etc. Work on a system which introduces these capabilities will be presented in a succeeding paper.

7 Acknowledgments

Much of the material in this paper is based on work initially expressed by the author in Chapters 4 and 5 of the *Handbook of Neural Computing Applications* (Maren et al., 1990) [9]. The author wishes to thank Academic Press for permission to reproduce Figures 2 - 5, which were intially printed in the *Handbook*. While writing a preliminary draft of this work (although entirely independent of contracted efforts), the author received contractual support from the U.S. Department of Energy (DoE-DEFG-88ER12824), the

U.S. Navy (N66001-89-C-7603), and the U.S. Marine Corps (N60921-90-C-A375). Additionally, the author also was supported by Accurate Automation Internal Research and Development.

Work on the *Cortical Engine*, described in Sect. 5.3, was developed independent of efforts at Accurate Automation, while the author was Visiting Associate Professor with the Dept. of Nuclear Engineering, University of Tennessee (Knoxville, TN), and was independent of support from any corporation or organization.

The author thanks Dr. Daniel C. Dennett for his helpful conversations (including a preprint of his 1987 manuscript) and for permission to reproduce Figure 1. Ms. Elizabeth McGowan prepared the figures used in this manuscript, and Ms. Vera Trotter assisted in typing.

References

[1] D. Dennett, "The logical geography of computational approaches (A view from the East Pole)," in *Problems in the Representation of Knowledge* (M. Harnish and M. Brand, eds.), Univ. Arizona Press, 1987.

[2] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, eds., *Parallel Distributed Processing*, vol. 1 and 2. Cambridge, MA: MIT Press, 1986.

[3] R. P. Lippmann, "Introduction to computing with neural networks," *IEEE ASSP Magazine*, pp. 4–22, April 1987.

[4] R. Hecht-Nielson, "Neurocomputing: Picking the human brain," *IEEE Spectrum*, pp. 36–41, March 1986.

[5] P. K. Simpson, "A review of artificial neural systems, I: Foundations, and II: Paradigms, applications, and implementation." Preprint privately circulated, published as book in 1990, 1988.

[6] P. K. Simpson, *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations.* New York: Pergamon, 1990.

[7] B. Kosko, "Unsupervised learning in noise," in *Proc. First Int'l. Joint Conf. Neural Networks*, vol. I, (Washington, DC), pp. I–7 – I–17, June 18-22 1989.

[8] D. W. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: An A/D Converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits and Systems*, vol. CAS-33, pp. 533–541, 1986.

[9] A. J. Maren, C. Harston, and R. Pap, *Handbook for Neural Computing Applications*. Academic Press, 1990. doi:10.13140/2.1.2917.5364.

[10] A. Lapedes and R. Farber, "Programming a massively paralle, computation universal system: Static behavior," in *Proc. IEEE Second Conf. on Neural Networks* (D. Anderson, ed.), (Snowbird, CA), pp. 283–298, 1986.

[11] P. J. Werbos, "Neurocontrol and related techniques," in *Handbook of Neural Computing Applications* (R. P. A. J. Maren, C. Harston, ed.), New York: Academic, 1990.

[12] P. J. Werbos, "Neurocontrol and supervised learning: An overview and evaluation," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches* (D. White and D. Sofge, eds.), pp. 65–90, New York: Van Nostrand, 1992.

[13] G. Carpenter and S. Grossberg, "ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures," *Neural Networks*, vol. 3, pp. 129–152, 1990.

[14] M. Minsky and S. Papert, eds., *Perceptrons*. Cambridge, MA: MIT Press, 1969, reprinted in 1988.

[15] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas imminent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943. Reprinted in J.A. Anderson and E. Rosenfeld (Eds.), *Neurocomputing: Foundations of Research*, 18-28. Cambridge, MA: MIT.

[16] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 358–408, 1958. Reprinted in J.A. Anderson and E. Rosenfeld (Eds.), *Neurocomputing: Foundations of Research*, 92-114. Cambridge, MA: MIT.

[17] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, Part 4*, pp. 96–104, 1960. Reprinted in J.A. Anderson and E. Rosenfeld (Eds.), *Neurocomputing: Foundations of Research*. Cambridge, MA: MIT.

[18] J. W. Clark, "Introduction to neural networks," in *Nonlinear Phenomena in Complex Systems* (A. Proto, ed.), pp. 1–102, Amsterdam: North-Holland, 1989.

[19] W. A. Little, "The existence of persistent states in the brain," *Math. Biosci.*, no. 19, pp. 101–120, 1974.

[20] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. USA*, vol. 79, p. 2554 2558, April 1982. Reprinted in J.A. Anderson and E. Rosenfeld (Eds.), *Neurocomputing: Foundations of Research*, 460-464. Cambridge, MA: MIT.

[21] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci. USA*, vol. 81, pp. 1088–1092, 1984. Reprinted in J.A. Anderson and E. Rosenfeld (Eds.), *Neurocomputing: Foundations of Research*, 579-584. Cambridge, MA: MIT.

[22] J. A. Anderson, "A simple neural network generating interactive memory," *Mathematical Biosciences*, vol. 14, pp. 197–220, 1972.

[23] T. Kohonen, "Correlation matrix memories," *IEEE Trans. Computers*, vol. C-21, pp. 1153–159, 1972.

[24] S. Grossberg, "Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors," *Bio. Cybernetics*, vol. 23, pp. 121–134, 1976.

[25] S. Grossberg, "Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, illusions," *Bio. Cybernetics*, vol. 23, pp. 187–202, 1976.

[26] G. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, & Image Processing*, vol. 37, pp. 54–115, 1987.

[27] G. Carpenter and S. Grossberg, "ART-2: Self-organization of stable category representation codes for analog input patterns," *Applied Optics*, vol. 24, pp. 4919–4930, 1987.

[28] R. P. Lippmann, B. Gold, and M. L. Malpass, "A comparison of Hamming and Hopfield neural networks for pattern classification," Tech. Rep. 769, MIT Lincoln Lab, Lexington, MA, May 21 1987.

[29] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable interpolation, and adaptive networks," Memo 4148, Royal Signals and Radar Establishment, Worchestershire, U.K., 1988.

[30] Y. H. Pao, *Adaptive pattern recognition and neural networks.* Boston, MA: Wesley Longman Publishing Co., Inc., 1989.

[31] P. J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* Ph.d. dissertation, Harvard University, Cambridge, MA, 1974.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing* (D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, eds.), vol. 1, pp. 318–362, Cambridge, MA: MIT Press, 1986.

[33] J. A. Anderson, J. W. Silverstein, S. A. Ritz, and R. S. Jones, "Distinctive features, categorical perception and probability learning: some applications of a neural model," *Psychological Review*, vol. 85, pp. 597–603, 1978.

[34] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, "The capacity of the Hopfield associative memory," *IEEE Trans. Information Theory*, vol. IT-33, pp. 461–482, 1987.

[35] T. Kohonen, *Self Organization and Associative Memory.* Berlin: Springer-Verlag, third ed., 1989.

[36] T. Kohonen, “The self-organizing map,” *Proc. IEEE*, vol. 78, pp. 1464 – 1480, 1990.

[37] S. Grossberg and E. Mingolla, “Neural dynamics of form perception: Boundary completion, illusory figures, and neoon color spreading,” *Psych. Review*, vol. 92, pp. 173–211, 1985.

[38] S. Grossberg and E. Mingolla, “Neural dynamics of perceptual grouping: Textures, boundaries, and emergent segmentation,” *Perception and Pyschophysics*, vol. 38, pp. 141–171, 1985.

[39] K. Fukushima, “Neocognitron: A hierarchical neural network capable of visual pattern recognition,” *Neural Networks*, vol. 1, no. 2, pp. 119–130, 1988.

[40] D. E. Rumelhart and D. Zipser, “Feature discovery by competitive learning,” in *Parallel Distributed Processing* (D. Rumelhart, J. McClelland, and the PDP Research Group, eds.), vol. 1, pp. 151–193, Cambridge, MA: MIT Press, 1986.

[41] A. J. Maren, C. T. Harston, and R. M. Pap, “A hierarchical structure approach for fusing multisensor information,” in *Proc. Second Nat'l. Symposium on Sensors and Sensor Fusion*, (Orlando, FL), March 27-31 1989. Distribution limited.

[42] A. J. Maren, V. Minsky, and M. Ali, “A multilayer cooperative/competitive method for creating hierarchical structures by clustering maximally-related nodes,” in *Proc. 19888 IEEE International Conference on Neural Networks*, pp. 95–105, 1988.

[43] A. J. Maren and V. Minsky, “Representing the perceptual organization of segmented images using hierarchical scene structures,” *J. Neural Network Computing*, vol. 1, pp. 14–33, Winter 1989. doi:TBD.

[44] D. M. Ackley, G. E. Hinton, and T. J. Sejnowski, “A learning algorithm for Boltzmann machines,” *Cognitive Science*, vol. 9, pp. 147–169, 1985. doi:10.1016/S0364-0213(85)80012-4.

[45] G. E. Hinton and T. J. Sejnowski, “Learning and relearning in Boltzmann machines,” in *Parallel Distributed Processing* (D. E. Rumelhart,

J. L. McClelland, and the PDP Research Group, eds.), vol. 1, pp. 282–317, Cambridge, MA: MIT Press, 1986.

[46] H. Szu, "Fast simulated annealing," in *Neural Networks for Computing* (J. Denker, ed.), vol. 15, pp. 420–425, AIP Conference, 1987.

[47] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[48] M. A. Cohen and S. Grossberg, "Masking fields: a massively parallel neural architecture for learning, recognizing, and predicting multiple groupings of patterned data," *Applied Optics*, vol. 25, pp. 1866–1891, 1987.

[49] G. N. Reeke, O. Sporns, and G. M. Edelman, "Synthetic neural modeling: The "Darwin" series of recognition automata," *Proc. IEEE*, vol. 78, pp. 1498–1530, 1990.

[50] D. Kleinfeld and H. Sompolinsky, "Associative network models for central pattern generators," in *Methods in Neuronal Modeling* (C. Koch and I. Segev, eds.), Cambridge, MA: MIT Press, 1989.

[51] J. Cowan, "A statistical mechanics of nervous activity," in *Biology - Lectures on Mathematics in the Life Sciences* (M. Gerstenhaber, ed.), vol. 2, pp. 1–57, Providence, RI: American Mathematics Society, 1970.

[52] R. Kikuchi and S. G. Brush, "Improvement of the Cluster Variation Method," *J. Chem. Phys.*, vol. 47, p. 195, 1967.