

Neural Network and Its Application in IR

Qin He

Graduate School of Library and Information Science

University of Illinois at Urbana-Champaign

Spring, 1999

Abstract

This is a literature review on neural networks and related algorithms. It is aimed to get a general understanding on neural networks and find out the possible applications of these models in information retrieval (IR) systems.

Beginning with a preliminary definition and typical structure of neural networks, neural networks are studied with respect to their learning processes and architecture structures. A case study on some specific networks and related algorithms is followed. The applications of some neural network models and related algorithms in information retrieval systems are then investigated. Problems on applying neural network models into IR systems are finally summarized in the conclusion.

1. Introduction

Neural network is one of the important components in Artificial Intelligence (AI). It has been studied for many years in the hope of achieving human-like performance in many fields, such as speech and image recognition as well as information retrieval. To make the term 'neural network' used in this paper clear and to expand considerably on its content, it is useful to give a definition to this term, analyze the general structure of a neural network, and explore the advantages of neural network models first.

1.1 The Definition of Neural Network

In his book (1990), Miller have found that "Neural networks are also called neural nets, connectionist models, collective models, parallel distributed processing models, neuromorphic systems, and artificial neural networks by various researchers (p.1-4)." Similarly, in his article (1987), Lippmann states, "artificial neural net models or simple 'neural nets' go by many names such as connectionist models, parallel distributed processing models, and neuromorphic systems (p.4)." However, Doszkocs and his coworkers (1990) think connectionist models are more general than neural network models and "they include several related information processing approaches, such as artificial neural networks, spreading activation models, associative networks, and parallel distributed processing (p. 209)." In their mind, "early connectionist models were called neural network models because they literally tried to model networks of brain cells (neurons) (p. 212)". A neural network model (or neural model) as that term is used refers to a connectionist model that simulates the biophysical information processing occurring in the nervous system. So, even though connectionist models and neural network models have same meaning in some literature, we prefer to regard connectionist models as a more general concept and neural networks is a subgroup of it.

A preliminary definition of neural network is given by Kevin Gurney in his course package (1999) as follows.

A Neural Network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or

weights, obtained by a process of adaptation to, or learning from, a set of training patterns.

Hecht-Neilsen Neurocomputer Corporation provides the following definition for neural network:

A cognitive information processing structure based upon models of brain function.

In a more formal engineering context: a highly parallel dynamical system with the topology of a directed graph that can carry out information processing by means of its state response to continuous or initial input (as cited in Miller, 1990, p. 1-3).

In its most general form, a neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest; the network is usually implemented by using electronic components or is simulated in software on a digital computer. Haykin (1999) has offered the following definition of a neural network viewed as an adaptive machine:

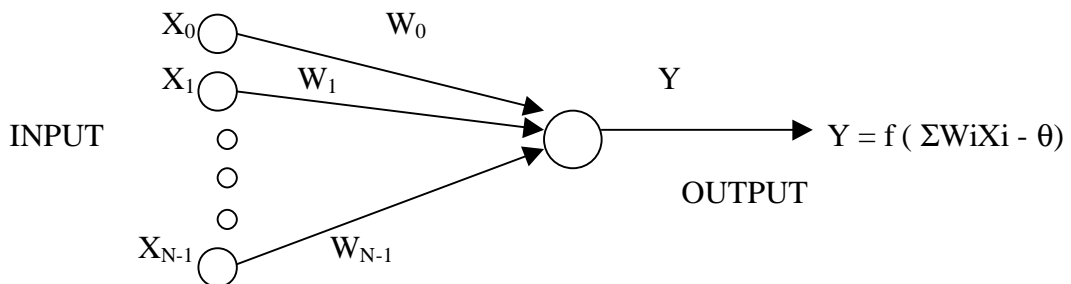
A neural network is a parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

- Knowledge is acquired by the network from its environment through a learning process;
- Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge (p. 2).

In summary, neural networks are models attempt to achieve good performance via dense interconnection of simple computational elements (Lippmann, 1987).

1.2 The Structure of Neural Networks

A simple neural network is given by Lippmann (1987, p. 5) as follows.



Considered as a typical connectionist model, Doszko, Reggia and Lin (1990) think a neural network has three components: a network, an activation rule, and a learning rule.

- The *network* consists of a set of nodes (units) connected together via directed links. Each node in the network has a numeric activation level associated with it at time t . The overall pattern vector of activation represents the current state of the network at time t .
- *Activation rule* is a local procedure that each node follows in updating its activation level in the context of input from neighboring nodes.
- *Learning rule* is a local procedure that describes how the weights on connections should be altered as a function of time.

Similarly, Haykin (1999) thinks there are three basic elements of the neuronal model, which include:

- A set of *synapses* or *connecting links*, each of which is characterized by a weight or strength of its own.
- An *adder* for summing the input signals, weighed by the respective synapses of the neuron. The operations could constitute a linear combiner.
- An *activation function* for limiting the amplitude of the output of a neuron. (p. 10)

The activation function is also referred to as a squashing function in that it squashes (limits) the permissible amplitude range of the output signal to some finite value. Types of activation functions include: 1) threshold function; 2) Piecewise-linear function, and 3) sigmoid function. The sigmoid function, whose graph is s-shaped graph, is by far the most common form of activation function used in the construction of neural networks (p. 14).

1.3. Historical Notes of Neural Network

Enthusiasm for and research activity on neural modeling waxed during the 1950s and 1960s. However, due both to methodological and hardware limitations and to excessively pessimistic attacks on the potential utility of these models, it was decreasing by the late 1960s. There is some important work on neural networks continued in the 1970s, but only a small number of researchers involved. During the 1980s, a

dramatically revived enthusiasm for neural networks occurred across a broad range of disciplines along with the appearances of new learning techniques and advances in the software and hardware of computer science and AI. This continues in the 1990s when more researchers are involved and new methods are proposed.

A short review of early neural network models is given by Doszkocs, Reggia and Lin (1990), with three representative examples as follows.

- *Networks based on logical neurons.* These are earliest neural network models. A logical neuron is a binary state device, which is either off or on. There is no mechanism for learning and a network for a desired input-output relationship must be designed manually.
- *Elementary Perceptron.* This is a kind of neural networks developed during the 1950s and 1960s, which learns through changes of synaptic strength. Given any set of input patterns and any desired classification of patterns, it is possible to construct an elementary perceptron that will perform the desired classification. The crowning achievement of work on elementary perceptrons is the perceptron convergence theorem.
- *Linear networks.* These are another class of neural models developed primarily during the 1960s and 1970s. Much work on linear networks has focused on associative memories.

In 1990, Widrow and Lehr (1990) reviewed the 30 years of adaptive neural networks. They gave a description of the history, origination, operating characteristics, and basic theory of several supervised neural network training algorithms including the perceptron rule, the LMS (least mean square) algorithm, three Madaline rules, and the back propagation techniques.

In his book, Haykin (1999) has provided some historical notes on neural networks based on a year by year literature review, which including McCulloch and Pitts's logical neural network, Wiener's *Cybernetics*, Hebb's *The Organization of Behavior*, and so on. As a conclusion, the author claims, "perhaps more than any other publication, the 1982 paper by Hopfield and the 1986 two-volume book by Rumelhart and McLelland were the most influential publications responsible for the resurgence of interest in neural network in the 1980s (p. 44)."

1.3 Advantages of Neural Network Models over Traditional IR Models

In neural network models, information is represented as a network of weighted, interconnected nodes. In contrast to traditional information processing methods, neural network models are "self-processing" in that no external program operates on the network: the network literally processes itself, with "intelligent behavior" emerging from the local interactions that occur concurrently between the numerous network components (Reggia & Sutton, 1988).

According to Doszkocs, Riggia and Lin (1990), neural network models in general are fundamentally different from traditional information processing models in at least two ways.

- First they are self-processing. Traditional information processing models typically make use of a passive data structure, which is always manipulated by an active external process/procedure. In contrast, the nodes and links in a neural network are active processing agents. There is typically no external active agent that operates on them. "Intelligent behavior" is a global property of neural network models.
- Second, neural network models exhibit global system behaviors derived from concurrent local interactions on their numerous components. The external process that manipulated the underlying data structures in traditional IR models typically has global access to the entire network/rule set, and processing is strongly and explicitly sequentialized (Doszkocs, Reggia & Lin, 1990).

Pandya and Macy (1996) have summarized that neural networks are natural classifiers with significant and desirable characteristics, which include but no limit to the follows.

- Resistance to noise
- Tolerance to distorted images/patterns (ability to generalize)
- Superior ability to recognize partially occluded or degraded images
- Potential for parallel processing

In a more general sense, Haykin (1999) has specified the use of neural networks offers the following useful properties and capabilities:

- *Nonlinearity.* A neural network, made up of an interconnection of nonlinear neurons, is itself nonlinear. Nonlinearity is a highly important property, particularly if the underlying physical mechanism responsible for generation of the input signal is inherently nonlinear.
- *Input-Output Mapping.* The network learns from the examples by constructing an input-output mapping for the problem at hand. Such an approach brings to mind the study of nonparametric statistical inference.
- *Adaptivity.* Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment.
- *Evidential Response.* In the context of pattern classification, a neural network can be designed to provide information not only about which particular pattern to select, but also about the confidence in the decision made.
- *Contextual Information.* Every neuron in the network is potentially affected by the global activity of all other neurons in the network.
- *Fault Tolerance.* Its performance degrades gracefully under adverse operating conditions.
- *VLSI Implementability.*
- *Uniformity of Analysis and Design.*
- *Neurobiological Analogy.*

2. A General Study of Neural Networks

A classification of neural networks given by Lippmann (1987) firstly divides neural networks into those with binary valued inputs and those with continuous valued inputs. Below this, nets are divided between those trained with and without supervision. A further difference between nets is whether adaptive training is supported.

As Bose (1996) summarized, neural networks not only have different structures or topology but are also distinguished from one another by the way they learn, the manner in which computations are performed (rule-based, fuzzy, even nonalgorithmic), and the

component characteristics. Here, we are going to do a general study on neural networks with respect to their learning processes and architecture structures.

2.1 Learning Processes in Neural Networks

An important property of neural networks is their ability to learn from input data with or without a teacher. Learning has long been a central issue for researchers developing neural networks. It is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded.

There are many rules used in the learning processes of neural networks. Haykin (1999) has categorized them into the following five groups.

1. Error-correction learning. The goal is to minimize the cost to correct the errors. This leads to the well-known *delta rule* (or *Widrow-Hoff rule*), which is stated as the adjustment made to a synaptic weight of a neuron is proportional to the product of the error signal and the input signal of the synapse in question.
2. Memory-based learning. All algorithms in this category involve two essential ingredients: 1) criterion used for defining the local neighborhood of the test vector X , and 2) learning rule applied to the training examples in the local neighborhood of X .
3. Hebbian learning. These are the oldest and most famous of all learning rules.
4. Competitive learning.
5. Boltzmann memory-based learning.

The type of learning in neural networks is determined by the manner in which the parameter changes. It can happen with or without a teacher. Typically, the learning processes in neural networks are divided into three groups: supervised learning, unsupervised learning and reinforcement learning.

2.1.1 Supervised learning

In a supervised learning process, the adjustment of weights is done under the supervision of a teacher; that is, precise information about the desired or correct network output is available from a teacher when given a specific input pattern.

Error-correction is the most common form of supervised learning. *Error* is defined as the difference between the desired response and the actual response of the network. The delta rule is always used in supervised learning. Unfortunately, the delta rule is directly applicable only to two-layer feed-forward networks. It does not tell what the error signals should be for nodes in hidden layers, because the correct response to an input node is only given by the teacher for output nodes. A generalized delta rule, the *back propagation algorithm* (BPA), is then developed and it provides a way to calculate the gradient of the error function efficiently using the chain rule of differentiation and it minimizes the squared-error measure over all patterns just like the delta rule (Doszko, Reggia, & Lin, 1990).

2.1.2 Unsupervised learning

In unsupervised or *self-organized* learning, the network is not given any external indication as to what the correct responses should be nor whether the generated responses are right or wrong. It is simply exposed to the various input-output pairs and it learns by the environment, that is, by detecting regularities in the structure of input patterns.

So, unsupervised learning aims at finding a certain kind of regularity in the data represented by the exemplars. Roughly speaking, regularity means that much less data are actually required to approximately describe or classify the exemplars than the amount of data in exemplars. Examples exploiting data regularity include vector quantization for data compression and Karhunen-Loeve expansion (often referred to as *principal component analysis*) for dimension reduction (Bose, 1996, p. 193).

In unsupervised learning, a simple Hebbian rule (*correlation rule*) may be applied to calculate weight changes. Energy-minimizing networks provide a recent example of unsupervised learning that make interesting use of a two-phase learning method. Competitive learning rules is another class of learning rules used in unsupervised neural networks. *Adaptive resonance theory* (ART) combines competitive and Hebbian rules together and uses feedback from the output layer to the input layer to ensure a consistent categorization. In an ART system, connections run in both directions, from input to output nodes and vice versa. Competitive learning is used to change weights on connections from the input to the output layer in creating groupings of the input patterns.

Hebbian pattern-association learning is used to change weights on connections from the output to the input layer. As a result, an input pattern evokes a pattern on the output layer, which in turn projects the prototype of the winning group back onto the input layer (Doszkocs, Reggia, & Lin, 1990).

2.1.3 Reinforcement learning.

Reinforcement learning is somewhere between supervised learning, in which the system is provided with the desired output, and unsupervised learning, in which the system gets no feedback at all. In reinforcement learning the system receives a feedback that tells the system whether its output response is right or wrong, but no information on what the right output should be is provided.

In reinforcement learning, a random search component is necessary. Since there is no information on what the right output should be, the system must employ some random search strategy so that the space of plausible and rational choices is searched until a correct answer is found. Reinforcement learning is also usually involved in exploring a new environment when some knowledge (or subjective feeling) about the right response to environmental inputs is available. The system receives an input from the environment and produces an output as response. Subsequently, it receives a reward or a penalty from the environment and learns from the environment (Bose, 1996).

The difficulty with reinforcement learning stems not only from the possibility that precise information about the error is unavailable but also from the likelihood that reward or penalty may be implemented only after many action steps (Bose, 1996, p.201). Competitive learning can be modified to use reinforcement learning. When a weight vector wins a competition, a reinforcement signal indicates whether or not this is a desirable outcome. A special class of gradient descent learning methods has also been studied in which a reinforcement signal is a probabilistic measure of the network's performance (Doszkocs, Reggia, & Lin, 1990).

2.2 Architecture Structures of Neural Networks

Neural networks are not only different in their learning processes but also different in their structures or topology. Bose (1996) has broadly classified neural

networks into recurrent (involving feedback) and nonrecurrent (without feedback) ones. In a little more details, Haykin (1999) has divided the network architectures into the following three classes.

2.2.1 Single-layer perceptrons (feed forward networks)

The single-layer perceptrons was among the first and simplest learning machines that are trainable. In Haykin's book (1999), perceptron denotes the class of two-layer feed forward networks, 1) whose first-layer units have fixed function with fixed connection weights from the inputs, and 2) whose connection weights linking this first layer to the second layer of outputs are learnable.

The model of training in perceptrons is supervisory, because the steps in the algorithm involve the comparison of actual outputs with desired outputs associated with the set of training patterns. An input layer of source nodes can project onto an output layer of neurons, but not vice versa. The LMS algorithm can be used in the supervisory training.

2.2.2 Multi-layer perceptrons (feed forward networks)

Multi-layer feed forward structures are characterized by directed layered graphs and are the generalization of those earlier single layer structures (Bose, 1996). A typical multi-layer feed forward network consists of a set of sensory units that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input signal propagates through the network in a forward direction on a layer-by-layer basis. These neural networks are also commonly referred to as multi-layer perceptrons (MLPs) (Haykin, 1999, p. 156).

A multi-layer perceptron has three distinctive characteristics (Haykin, 1999):

1. The model of each neuron in the network includes a nonlinear activation function;
2. The network contains one or more layers of hidden neurons that are not part of the input or output of the network;
3. The network exhibits a high degree of connectivity, determined by the synapses of the network.

Multi-layer feed forward networks are always trained in supervised manner with a highly popular algorithm known as the error back propagation algorithm. Basically, error back propagation learning consist of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, an activity pattern (input vector) is applied to the sensory node of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual response of the network. The synaptic weights of the networks are all fixed during the forward pass. The backward pass starts at the output layer by passing error signals leftward through the network and recursively computing the local gradient for each neuron. This permits the synaptic weights of the network to be all adjusted in accordance with an error-correction rule (Haykin, 1999).

2.2.3 Recurrent networks

In the neural network literature, neural networks with one or more feedback loops are referred to as recurrent networks. A recurrent network distinguishes itself from a feed forward neural network in that it has at least one feedback loop. Such a system has very rich temporal and spatial behaviors, such as stable and unstable fixed points and limit cycles, and chaotic behaviors. These behaviors can be utilized to model certain cognitive functions, such as associative memory, unsupervised learning, self-organizing maps, and temporal reasoning.

Feedback plays a major role in the study of recurrent networks. There are two ways for feedback: local feedback at the level of a single neuron inside the network and global feedback encompassing the whole network.

Bose (1996) has pointed out that the problems that need to be addressed in the study of recurrent networks including the followings:

1. Network synthesis and learning algorithm.
2. Convergence and learnability
3. Size of the network
4. Complexity of analog computation

Bose (1996) has also divided recurrent networks into symmetric and asymmetric recurrent ones based the weights of their connections.

- **Symmetric Recurrent Network.** In symmetric recurrent network, the connections are symmetric, that is, the connection weights from unit i to unit j and from unit j to unit i are identical for all i and j . The widely known Hopfield networks, are a kind of symmetric recurrent networks. Symmetric networks always converge to stable point attractors. As such, a symmetric network cannot generate, learn, or store a temporal sequence of patterns. This leads to the study of the asymmetric networks.
- **Asymmetric Recurrent Network.** The dynamic behavior of asymmetric networks includes *limit cycles* and *chaos*, and these networks are capable of storing or generating temporal sequences of spatial patterns. Chaos in a recurrent neural network is characterized by a time evolution that progresses through a set of distorted patterns in a notably irregular manner. In a stationary state, on the other hand, the state from one time index to the next does not change. When the recurrent network cycles through a predetermined sequence of patterns, a limit cycle is said to occur. It has been shown that an asymmetric network with a large number of units may undergo a transition from stationary states to chaos through an intermediate stage of limit cycles with increasing complexity (p. 331).

Recurrent back propagation (RBP) is one way to extend the standard back propagation algorithm to recurrent networks (Bose, 1996). RBP is a nonrule-based continuous-time formalism that emphasizes the dynamics of the network for computation. Although an elegant idea, RBP faces many unanswered questions. In general, understanding of recurrent networks with asymmetric connections is limited. This is a reflection on the limited mathematical tools available for analyzing the dynamics of general nonlinear system.

3. Study of Special Cases and Related Algorithms

In this section, three classes of neural networks will be studied in details. These are multi-layer feed forward networks, Kohonen self-organizing feature maps (SOFM) and Hopfield networks. In addition, semantic networks will also be studied as they are related to neural networks.

3.1 Multi-layer Perceptron Networks and Back Propagation Algorithm

Multi-layer perceptron (MLP) networks trained by back propagating are among the most popular and versatile forms of neural network classifiers. This section will study the structure and features of MLP, the back propagation algorithm, and a variant of MLP.

3.1.1 Structure and features of MLP

Multi-layer perceptron (MLP) networks are feed forward nets with one or more layers of nodes between the input and output nodes. The structure of an unadorned multi-layer perceptron network is shown in figure 1.

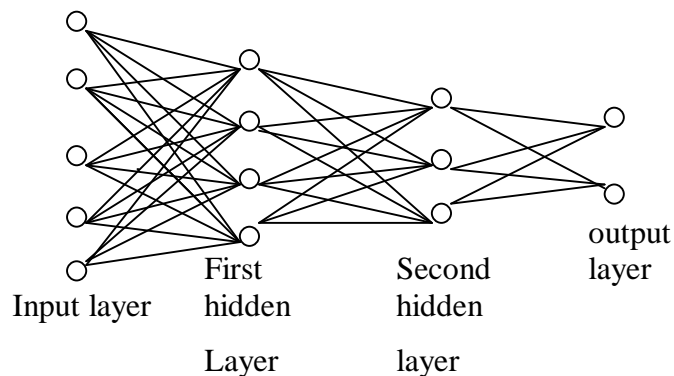


Figure 1. Feed forward multi-layer perceptron architecture
(Pandya & Macy, 1996, p.74)

The capabilities of multi-layer perception stem from the nonlinearities used within nodes. Compared to single-layer perceptron nets, Lippmann (1987) have got the following conclusions about MLP.

- No more than three layers are required in perceptron-like feed-forward nets because a three-layer net can generate arbitrarily complex decision regions.
- The number of nodes in the second layer must be greater than one when decision regions are disconnected or meshed and cannot be formed from one

convex areas. The number of second layer nodes required in the worst case is equal to the number of disconnected regions in input distributions.

- The number of nodes in the first layer must typically be sufficient to provide three or more edges for each convex area generated by every second-layer node (p. 16).

3.1.2 Back propagation algorithm

Back propagation is one of the simpler members of a family of training algorithms collectively termed gradient descent. The idea is to minimize the network total error by adjusting the weights. Gradient descent, sometimes known as the method of steepest descent, provides a means of doing this (Pandya & Macy, 1996). An outline of the back propagation training algorithm is given by Lippmann (1987) as follows. It requires continuous differentiable non-linearities.

Step 1. Initialize weights and offsets. Set all weights and node offsets to small random values.

Step 2. Present input and desired output. The desired output is 1. The input could be new on each trial or samples from a training set.

Step 3. Calculate actual outputs. Use the sigmoid nonlinearity formulas to calculate outputs.

Step 4. Adapt weights.

Step 5. Repeat by going to step 2.

It is important to make the initial weights “small”. Choosing initial weights too large will make the network untrainable. One difficulty with the backward-propagation algorithms is that in many cases the number of presentations of training data required for convergence has been large (Lippmann, 1987, p. 18). It has been shown that multi-layer perception networks with a single hidden layer and a nonlinear activation function are universal classifiers (Pandya & Macy, 1996, p.73).

There are many variants of BPA, such as the Newton-Raphson method or the conjugate gradient method. However, these variants also share problems present in the standard BPA and may converge faster in some cases and slower in others. Therefore, they may work better in some cases and worse in others (Bose, 1996p.178). Lack of any

convergence criterion is a severe drawback for error back propagation structures, especially when different classes of patterns are close to each other in multidimensional feature space.

Different approaches, such as *genetic algorithms*, exist that avoid use of any gradient information. The genetic algorithms provide a different approach to organize the search in parameter space so that there is a high likelihood of locating either an optimal or near-optimal solution. Unlike the usual back propagation, where the gradient descent methods and their variants iteratively refine a trial solution until no further improvements result, the genetic search, inspired by biological evolution, cross-breeds trial solutions by selective reproduction, recombination, and mutation, permitting only the fittest solutions to survive after several generations. Genetic algorithms have the ability to adapt to the problem being solved and are suggested by the evolutionary process of nature selection. Since no differentiation is required, neither the fitness function nor the transfer characteristics of the units need be differentiable. Therefore, TLUs (Threshold Logic Unit) can be used in stead of sigmoidal function units (Bose, 1996, p. 179).

3.1.3 Probabilistic network (PNN)

PNN is another kind of multi-layer feed forward network. In addition to the input layer, the PNN has two hidden layers and an output layer. The major difference from a feed forward network trained by back propagation is that it can be constructed after only a single pass of the training exemplars in its original form and two passes in a modified version. The activation function of a neuron in the case of PNN is statistically derived from estimates of probability density functions (PDFs) based on training patterns (Bose, 1996, p. 204).

The principal advantage of the PNN is fast construction that does not require an iterative procedure as the back propagation algorithm does. It can be effective in situations where there is sparse data in a real-time environment. When the size of the training exemplar set is large, various clustering techniques can be applied to generate a smaller set of representative exemplar. The main disadvantage is the amount of

computation required after training when a new output is to be calculated (Bose, 1996, p. 211-212).

3.2 Kohonen Networks and Learning Vector Quantization

From a simple Kohonen net to the complicated self-organizing feature maps (SOFM), Kohonen has been worked on this kind of neural networks for a long time (Kohonen, 1988). The architecture for a SOFM has the special property of being able to create effectively topographically organized maps of the different features of exemplar patterns. The key to the SOFM's spatial organization capability is the ability to select a winning neighborhood instead of a single winner. LVQ (learning vector quantization) also introduced by Kohonen is shown to be useful for clustering abstract pattern vectors in data compression.

3.2.1 A simple Kohonen net

A simple Kohonen net architecture consists of two layers, an input layer and a Kohonen (output) layer. These two layers are fully connected. Each input layer neuron has a feed-forward connection to each output layer neuron (see figure 2).

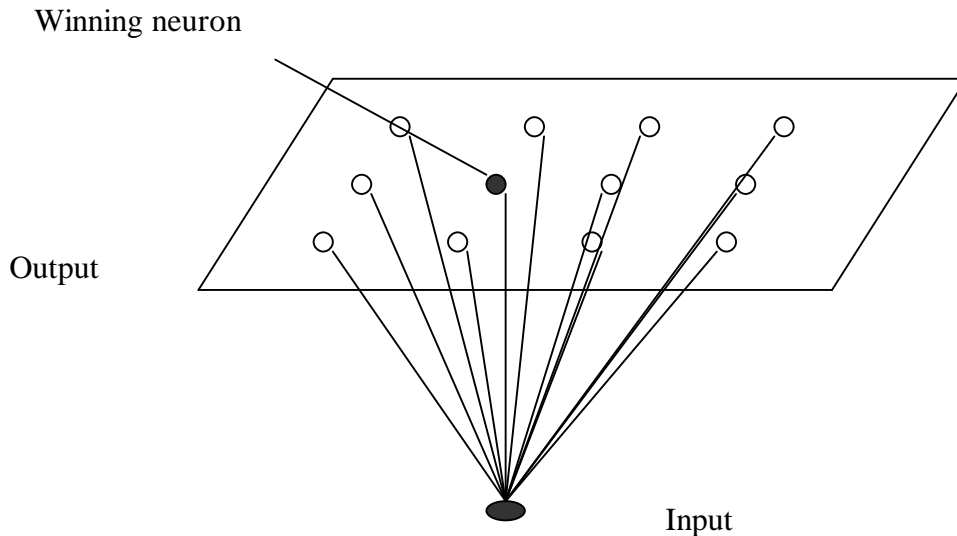


Figure 2. Kohonen model (Haykin, 1999, p. 445)

A Kohonen network works in two steps. First, it selects the unit whose connection weight vector is closest to the current input vector as the winning unit. After a winning neighborhood is selected, the connection vectors to the units whose output values are positive are rotated toward the input vector.

Inputs to the Kohonen layer (i.e., the output layer) can be calculated by the dot product between a neuron weight vector and the input vector. The winning output layer neuron is simply the neuron with the biggest dot product. All other neurons in the Kohonen layer output nothing. The angle between the winning neuron weight vector and the input vector is smaller than the other neurons.

An alternative method of choosing the winning neuron simply selects the neuron whose weight vector has a minimum of the Euclidean norm distance from the input vector. For vectors of unit length this method is equivalent to the method just described in that the same neuron will be chosen as the winner. The use of Euclidean distance to select a winner, however, may be advantageous in that it does not require weights or input vectors to be normalized (Pandya & Macy, 1996, p. 233).

3.2.2 SOFM and competitive learning

The goal of SOFM is the mapping of an input space of n-dimensions into a one or two- dimensional lattice (of output layer neurons) which comprises the output space such that a meaningful topological ordering exists within the output space. Such a map is called a topographic map (Bose, 1996, p. 361).

SOFM networks are distinguished by their use of competitive (unsupervised) learning. Competitive learning can be described as a process in which output layer neurons compete among themselves to acquire the ability to fire in response to given input patterns. A winner-take-all CLN (competitive learning network) consists of an input layer and a competition, or output layer. The input layer is connected fully to the output layer with feed forward connections only (Bose, 1996). A problem with competitive learning is stability. For complex environments the weight vectors can continually drift, causing an inconsistent categorization of the input patterns (Doszkocs, Reggia, & Lin, 1990).

Competitive learning involves two phrases. The first phrase is the competition phase, in which a winner is selected. The second phase is the reward phase, in which the winner is rewarded with an update of its weights. Generally, selection of the winner occurs at the output layer. Updating of the weights occurs on the connections from the input layer to the output layer (Bose, 1996).

One application of CLN is in associative recall. In this application, instead of learning the connection weight vector, the *Hamming distances* (HDs) between the connection weight vectors and the input vector are computed. The connection weight vector associated with the smallest HD to the input vector determines the recall result. A CLN used for this purpose is called a *Hamming network* (HN) (Bose, 1996, p. 349-350). The Hamming network implements an optimal minimum error classifier and scales more favorably than a Hopfield network.

3.2.3. Learning vector quantization (LVQ)

Since the size of the vector in a SOFM can be very large in some cases, transmission and digital processing of the data usually done after compression. Vector quantization (VQ) is often used for data compression. VQ may be viewed as a data-reducing technique, where all input vector that fall into particular cell are represented by a much smaller code word associated with that cell (Bose, 1996).

Kohonen extended the so-called adaptive vector quantization (AVQ) algorithms using his unsupervised competitive learning rule to several learning vector quantization (LVQ) algorithms what based on supervised learning rules (Bose, 1996, p. 375). These LVQ algorithms proposed by Kohonen are for fine-tuning the map after its initial ordering. These allow specification of the categories into which inputs will be classified. The designated categories for the training set are known in advance and are part of the training set. LVQ network architecture is exactly the same one as the SOFM with the single exception that each neuron in the output layer is designated as belonging to one of the several classification categories (Pandya & Macy, 1996, p. 259).

3.3 Associative Memories and the Hopfield Net

A memory system that can recall a pattern from memory based on the contents of an input is an associative or content-addressable memory. If pattern is recalled from memory when a sufficiently similar pattern is presented, the memory is autoassociative. The other type of associative memory, in which the input and output vectors may have entirely different connotations, is a heteroassociative memory. The Hopfield network is an example of autoassociative memory with feedback. The basic model consist of a set of processing elements that compute the weighted sums of the inputs and threshold the outputs to binary values (Bose, 1996).

3.3.1 Hopfield network

Consider a network in which every unit is connected to every other unit and the connections are symmetric. Such a network is called a Hopfield network.

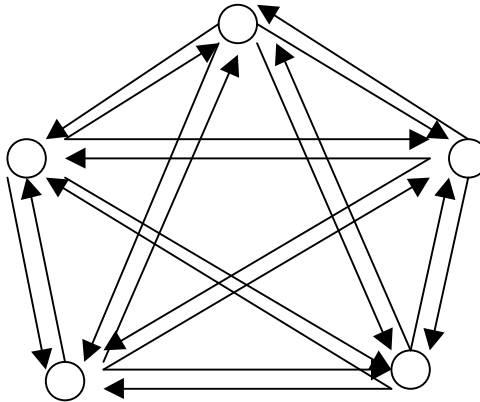


Figure 4. The structure of a Hopfield network (Bose, 1996, p. 290)

Lippmann (1987) summarized the operation of the nets as follows. First, weights are set using the given recipe from exemplar patterns for all classes. Then an unknown pattern is imposed on the net at time zero by forcing the output of the net to match the unknown pattrer. Following this initialization, the net iterates in discrete time steps using the given formula until it is converged, that is, outputs no longer change on successive iterations.

A sketch of the Hopfield net activation algorithm follows:

1. Assigning synaptic weights;

2. Initialization with search terms;
3. Activation, weight computation, and iteration;
4. Convergence.

The Hopfield net has two major limitations when used as a content addressable memory. First, the number of patterns can be stored and accurately recalled is severely limited. Second, an exemplar pattern will be unstable if it shares many bits in common with another exemplar pattern (Lippmann, 1987).

3.3.2 Simulated annealing and the variants of Hopfield Nets

A Hopfield network follows a gradient descent rule. Once it reaches a local minimum, it is stuck there. To find the global minimum of a function using only local information, some randomness must be added to the gradient descent rule to increase the chance of hitting the global minimum.

Simulated annealing (SA) is a method that introduces randomness to allow the system to jump out of a local minimum. The method draws an analogy between optimization of a cost function in a large number of variables with minimization of an energy function in statistical mechanics (Bose, 1996, p. 319).

The idea of simulated annealing can be applied to search for the global minimum in a discrete Hopfield network. Applying simulated annealing to Hopfield networks results in a variant of Hopfield net called *Stochastic Networks* (Bose, 1996, p. 323).

The simulated annealing method of optimization also leads to the *Boltzmann machine* (BM). Boltzmann Machine is another possible extension of the Hopfield network obtained by introducing hidden units. The state of these hidden units can acquire an internal representation of the input patterns. The distinguishing feature is that the learning rule for such a machine is not deterministic but stochastic. The learning algorithm is based on the Boltzmann-Gibbs distribution at thermal equilibrium and is developed for networks with hidden units and symmetric connections. The BM, being a generalization of the Hopfield network, can serve also as a heteroassociative memory in addition to an autoassociative memory (Bose, 1996, p. 324).

3.4 Semantic Networks

Semantic networks is another class of connectionist models emerged in cognitive psychology during the 1960s and 1970s. Strictly speaking, they are not neural networks. However, they have similar architectures as neural networks and they work in a similar way to neural networks.

Semantic networks typically had nodes that represented concepts and connections that represented semantically meaningful associations between these concepts. Thus, they are better characterized as associative network models than as neural/brain models. The activation rules that implement information retrieval in these associative networks, often referred to as spreading activation, typically produces an intersection search. So, they are also called "spreading activation" models (Doszko, Reggia & Lin, 1990).

In semantic networks, the conceptual units represented by nodes are semantic entities (e.g., lamp, attic, liver disease, jaundice), and the relationships represented by connections are semantic associations -- e.g., the propositions "location (lamp, attic)" and "causes (liver disease, jaundice)" might each be represented by a link between the appropriate nodes. Semantic networks have been used widely in traditional AI problem-solving systems, but as these systems have become more powerful, they have become progressively larger and slower (Doszko, Reggia & Lin, 1990).

Many systems for representing knowledge can be considered semantic networks largely because they feature the notion of an explicit taxonomic hierarchy, a tree or lattice-like structure for categorizing classes of things in the world being represented. There are some sort of "inheritance" links between the representational objects and these links called "IS_A" have been perhaps the most stable element of semantic nets as they have evolved over the years. Unfortunately, this stability may be illusory. There are almost as many meanings for the IS-A links as there are knowledge-representation systems. Brachman (1983) has studied the IS-A semantic network in details and found that there are two types of IS-A relations, that is, generic/generic relations and generic/individual relations. It might be much clearer if IS-A were broken down into its semantic sub-components, and those sub-components were then used as the primitives of a representation system.

4. Application of Neural Network Models in IR

Neural networks computing, in particular, seems to fit well with conventional retrieval models such as the vector space model and the probabilistic model. Doszkocs et al. (1990) provided an excellent overview of the use of connectionist models in IR. A major portion of research in IR may be viewed within the framework of connectionist models. For example, to the extent that all connectionist models can be regarded as input-to-output classificatory systems, document clustering can be viewed as classification in the document*document space. Thesaurus construction can be viewed as laying out a coordinate system in the index*index space. Indexing itself can be viewed as mappings in the document*index space, and searching can be conceptualized as connections and activations in the index*document space.

Kinoshita and Palevsky (1987) predict that applying connectionist approaches to information retrieval will likely produce information systems that will be able to:

- 1) recall memories despite failed individual memory units;
- 2) modify stored information in response to new inputs from the user;
- 3) retrieve "nearest neighbor" data when no exact data match exists;
- 4) associatively recall information despite noise or missing pieces in the input, and
- 5) categorize information by their associative patterns.

In this section, we will study the applications of some neural network models (i.e., SOFM and Hopfield network) and related algorithms (i. e., semantic network) in information retrieval systems.

4.1 The Application of SOFM

Most of the applications of SOFM in IR systems are base on the fact that SOFM is a topographic map and can do mappings from a multidimensional space to a two- or three- dimensional space.

Kohonen has shown that his self-organizing feature map "is able to represent rather complicated hierarchical relations of high-dimensional space in a two-dimensional display (Kohonen, 1988, p. 141)." He gives two examples related to information

retrieval. In the first example, he uses his feature map and a conventional hierarchical clustering analysis to process some artificial data that can be viewed as a matrix of 32 documents and five indexing terms. Both methods resulted in similar minimal spanning trees that maintained the same topological relationships between neighbors. In his second example, he showed that the self-organizing mapping could also display, two-dimensionally, topological and similarity relations between phonemes from continuous speech. He had concluded that "the self-organized mappings might be used to visualize topologies and hierarchical structures of high-dimensional pattern spaces (Kohonen, 1988, p. 142)." Document space is certainly such a high-dimensional space.

Hatano and his colleagues (1997) have proposed an information organizer for effective clustering and similarity-based retrieval of text and video data using Kohonen's self-organizing map. In their model, a vector space model and DCT (Discrete Cosine Transform) image coding is used to extract characteristics of data and SOFM is used to cluster data. There are two notable features of this organizer: content-based clustering by SOFM and retrieval and navigation on 3D overview map. A comparison between the word-frequency based algorithms of SOFM and algorithms based on Salton's measurement shows the former seems to be more suitable to cluster documents and to generate a global overview map while the latter seems to be more effective to perceive each document's distinction which is useful to information retrieval (p.210).

SOFM is also used by the DLI (digital library initiative) project at the University of Illinois. On one hand, it is used to classify and map the category of text documents. On the other hand, it is used to map the category of the texture of images (Pape, 1998). Similarly, Soergel & Marchionini (1991) adopted Kohonen networks to construct a self-organizing (unsupervised learning), visual representation of the semantic relationships between input documents (as cited in Chen, 1995).

4.2. Application of Hopfield Net

Hopfield net was introduced as a neural net that can be used as a content-addressable memory. Knowledge and information can be stored in single-layered interconnected neurons (nodes) and weighted synapses (links) can be retrieved based on

the network's parallel relaxation method. It had been used for various classification tasks and global optimization.

A variant of Hopfield network is developed by Chen and his colleagues (Chen, et al, 1993) to create a network of related keywords. It uses an asymmetric similarity function to produce thesauri (or knowledge bases) for different domain-specific databases. These automatic thesauri are then integrated with some existing manually created thesauri for assisting concept exploration and query refinement. In addition, a variant of the Hopfield parallel relaxation procedure for network search and concept clustering is also implemented (as cited in Chen, 1995).

In Chung and her colleagues' study (Chung, et. al, 1998), the Hopfield network has been adapted for the special needs of information retrieval -- concept assigner. The network is an asymmetric, continuous network in which the neurons are updated synchronously. The major steps of the algorithm are:

- 1) Assigning synaptic weights. The Concept space generated by similarity analysis serves as a trained network in the system. The concepts in the Concept space represent nodes in the network and the similarities, computed based on co-occurrence analysis, represent synaptic weights between nodes (concepts).
- 2) Initialization. An initial set of concept (noun phrases) extracted from a document serves as the input pattern.
- 3) Activation. Nodes in the concept space Hopfield net are activated in parallel, and activated values from neighboring nodes are combined for each individual node.
- 4) Convergence. The above process is repeated until the network reaches a stable state, i.e., there is no significant change in the value of the output states between two time steps.

4.3 The Applications of MLP Networks and Semantic Networks

It is hard to distinguish the applications of MLP and the applications of semantic networks with spreading activation methods in IR. In most cases, the applications of semantic networks in IR are making use of spreading activation models while having a feed-forward network structure similar to that of MLP networks.

Wong and his colleagues (1993) have developed a method for computing term associations using a three-layer feed-forward network with linear threshold functions. Each document is represented as a node in input layer. The nodes in the hidden layer represent query terms and the output layer consists of just one node, which pools the input from all the query terms. Term associations are modeled by weighted links connecting different neurons, and are derived by the perceptron learning algorithm without the need for introducing any ad hoc parameters. The preliminary results indicate the usefulness of neural networks in the design of adaptive information retrieval systems.

Wikinson and Hingston (1991) implemented a document retrieval system based on a neural network model. Since neural networks can perform very well at matching a given pattern against a large number of possible templates. They used this organization for selecting relevant documents. There is a 3-layer network in this system. One layer for query terms, one layer for terms in all documents and one layer for all the documents. Based on term frequency and document frequency, query terms activate terms in document and rank the relevant documents. The advantages of this ranking system include: 1) It allows for standard document ranking, as determined by cosine measure; 2) it allows the system to find words that appear to be relevant on the basis of initial ranking, and use those words to refine the document ranking; 3) it allows for a simple incorporation of relevance feedback in the vector space model. It was shown that a neural net structure can be used for ranking documents in a flexible fashion that allows for a variety of inputs to influence the final ranking and many of the standard strategies of information retrieval are applicable in a neural network model.

Similarly, in Kwok's work, he also represented an IR system into a 3-layer network -- queries connected to index terms to documents. He attempted to employ the neural network paradigm to reformulate the probabilistic model of IR with single term as document components. A probabilistic retrieval system is aimed to provide an optimal ranking of a document collection with respect to a query. This is based on a decision function, which maybe a variant of Bayes' Theorem. In this 3-layer network, the discrimination function between neurons is still based on inverse-document-frequency (IDF), but there are learning algorithms exist. It is proved that the activation through

network can provide much better results than the traditional, document-mode IDF weighting.

In his article, Belew (1989) talked about adaptive information retrieval (AIR) which represents a connectionist approach to the task of information retrieval. The system uses relevance feedback from its users to change its representation of authors, index terms and documents so that, over time, it improves at its task. The author argued that connectionist representations are particularly appropriate for IR for two reasons. First these networks naturally perform a type of spreading activation search that is shown to be a natural extension of techniques used in IR systems. Second, powerful learning algorithms have been developed for connectionist systems that allow these representations to improve over time. This offers the potential of IR systems that automatically modify their indices to improve the probability of relevant retrievals (p. 11). Generalized representation, generalized retrieval, and generalized input and output are the distinguish features on AIR systems.

Strictly speaking, these networks are semantic networks and work in the spreading activation models. Similar works include Preece (1981), Cohen & Kheldsen (1987) and so on. These examples show the potential of spreading activation for information retrieval. In theory, because of its inherent associative approach, spreading activation is believed to have the potential to outperform "exact-match" techniques. However, experimental verification and comparative evaluation are still needed. In fact, the comparison done by Salton & Buckley (1988) of a spreading activation model and vector processing methods questions this belief. These authors compared four variants of the spreading activation model of Jones & Furnas with four variants of vector processing methods in six different databases. The results indicate that vector processing methods produce better results than the spreading activation methods. Thus, these authors concluded that the simple spreading activation model they considered "may not be sufficiently powerful to produce acceptable retrieval output". Perhaps one insight from Salton & Buckley's comparison is that successful spreading activation may need to be based on a well-learned network. An obvious disadvantage of spreading activation is that control of spreading activation relies on parameter selection (Doszkoacs, Reggia, & Lin, 1990, p.236).

5. Conclusions

The literature review above shows neural network models have many attracting properties and some of them could be applied into an IR system. It is expected the research on the application of neural network models into IR will grow rapidly in the future along with the development of its technological basis both in terms of hardware and software. Computer hardware in the future will be more suitable for supporting "massive" explicit parallelism of the sort used in neural network models. Software developments can also be expected, not only in terms of new methods for controlling network functionality but also in terms of the availability of general-purpose software environments for implementing and using neural network models, such as some application packages of neural networks based on MATLAB (<http://www.mathworks.com/products/neuralnet/>).

However, there are still some unsolved questions found by Doszkocs, Reggia and Lin (1990) when they study connectionist models, which include:

- 1) Which connectionist techniques are appropriate for which information retrieval problems?
- 2) How should we compare connectionist approaches with traditional information retrieval techniques, and what is the best way to evaluate their performance?
- 3) How do we overcome problems of scale limitations of connectionist implementations in real-life, large-scale operational systems?
- 4) Is it desirable and productive to look for new connectionist paradigms for information retrieval, or should research be aimed at finding better symbolic representations of document retrieval activities by people? (p. 243)

As a class of connectionist models, the study of neural network models is also facing similar problems to these. None of these questions is trivial. We can not envision that neural network models will soon replace traditional IR approaches. However, we believe the application of neural network models can make an IR system more powerful.

Acknowledgements:

The report is finished under the instruction of Professor Linda C. Smith. The student is grateful to her for her help in the whole process. Special thanks also to all the other members of the student's advisory committee, Professor Bruce R. Schatz, P. Bryan Heidorn, and David S. Dubin, for providing the reading list and helpful comments.

References:

- Belew, R. K. (1989). Adaptive information retrieval: using a connectionist representation to retrieve and learn about documents. *ACM SIGIR'89*, 11-20.
- Bose, N. K.; & Liang P. (1996). *Neural network fundamentals with graphs, algorithms, and applications*. McGraw-Hill, Inc.
- Brachman, R. J. (1983). What is-a is and isn't: an analysis of taxonomic links in semantic networks. *IEEE Computer*, Oct. 30-36.
- Chen, H.; et al. (1993) Generating, integrating, and activating thesauri for concept-based document retrieval. *IEEE Expert*, 8, 25-34.
- Chen, H. (1995). Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithm. *Journal of the American Society for Information Science*, 46(3): 194-216.
- Chung, Y-M. Potternger, W.M.; & Schatz, B. R. (1998). Automatic subject indexing using an associative neural network. In Ian Witten, Rob Akscyn; & Frank M. Shipman, III (eds.). *Digital Libraries 98/ The 3rd ACM conference on digital libraries*, 59-68.
- Cohen, P. R.; & Kjeldsen, R. (1987). Information retrieval by constrained spreading activation in semantic networks. *Information Processing and Management*, 23(4), 255-268.
- Doszkocs, T. E.; Reggia, J.; & Lin, X. (1990). Connectionist models and information retrieval. *Annual Review of Information Science and Technology (ARIST)*, 25, 209-260.
- Grossberg, S. (1976). Adaptive pattern classification and universal recording: 1. parallel development and coding of neural feature detectors. *Biological Cybernetics*, 23:121 - 134.

- Gurney, K. (1999). Neural Nets by Kevin Gurney.
<http://www.shef.ac.uk/psychology/gurney/notes/index.html>, visited on Jan. 3rd.
- Hatano, K.; Qian, Q.; & Tanaka, K. (1997). A SOM-based information organizer for text and video data. Proceedings of the Fifth International Conference on Database Systems for Advanced Applications, Melbourne, Australia, April 1-4, 205-214.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, USA, vol. 84, pp.8429-8433
- Hopfield, J. J. and Tank, D. W. (1986). Computing with neural circuits: A model. Science, 233:625 -633.
- Haykin, S. (1999). Neural networks: a comprehensive fundation. (2nd ed.) Upper Saddle River, New Jersey: Prentice Hall
- Kinoshita, J.; & Palevsky, N. G. (1987). Computing with neural networks. High Technology, 7(5), 24-32.
- Kohonen, T. (1988) Self-Organization and Associative Memory. 2nd Edition. Berlin: Springer-Verlag.
- Kwok, K. L. (1989). A neural network for probabilistic information retrieval. ACM SIGIR'89, 21-30.
- Lippmann, R. P. (1987). An introduction to computing with neural nets. IEEE ASSP Magazine, 4-22.
- McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 7:115 - 133.
- Miller, R. K. (1990). Neural Networks. Lilburn, GA: Fairmont Press, Inc.
- Pandya, A. S. and Macy, R. B. (1996). Pattern Recognition with Neural Networks in C++. CRC Press, Inc.
- Pape, D. X. (1998). <http://www.canis.uiuc.edu/interspace/showcase.html>.
- Preece, S. E. (1981). A spreading activation network model for information retrieval. PhD thesis, Computer Science Department, University of Illinois, Urbana, IL, 1981.
- Reggia, J. A.; & Sutton, G. G., III. (1988). Self-processing networks and their biomedical implications. Processings of the IEEE, 76, 680-692.

- Rumelhart, D. E. and McClelland, J. L. (eds.). (1986). *Parallel distributed processing: explorations in the microstructure of cognition*. Cambridge, MA: MIT press.
- Salton, G; & Buckley, C. (1988). On the use of spreading activation methods in automatic information retrieval. In: Chiaramella, Y. (ed.). 11th International Conference on Research & Development in Information Retrieval, June 13-15, Grenoble, France. New York, NY: Association for Computing Machinery, 147-160.
- Von der Malsburg, C. (1973). Self-organisation of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85 - 100.
- Wikinson, R.; & Hingston, P. (1991). Using the cosine measure in a neural network for document retrieval. *ACM SIGIR'91*, 202-210.
- Wong, S. K. M.; Cai, Y. J.; & Yao, Y. Y. (1993). Computation of term associations by a neural network. *ACM-SIGIR'93*, 107-115.