Neural Network Classification and Formalization

E. Fiesler

IDIAP
Case postale 609
CH-1920 Martigny
Switzerland

Abstract

In order to assist the field of neural networks in maturing, a formalization and a solid foundation are essential. Additionally, to permit the introduction of formal proofs, it is essential to have an all-encompassing formal mathematical definition of a neural network.

This publication offers a neural network formalization consisting of a topological taxonomy, a uniform nomenclature, and an accompanying consistent mnemonic notation. Supported by this formalization, both a flexible hierarchical and a universal mathematical definition are presented.

Keywords: (artificial) neural network, neural computing, neurocomputing, connectionism, formalization, standardization, terminology, nomenclature, definition, mnemonic notation, topological taxonomy, neural network classification, neural network determination

[0]This document has been accepted for publication in: *Computer Standards & Interfaces*, volume 16, special issue on Neural Network Standards, John Fulcher, editor, North-Holland, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 1994, ISSN 0920-5489.

1 Introduction

The field of (artificial) neural networks has evolved independently in and among many communities such as neurobiology, mathematics, computer science (artificial intelligence), physics, and psychology. As a result, a variegated terminology and synonymic nomenclature developed. Coincidingly, there has been a parallel evolution of neural network definitions. This is reminiscent of artificial intelligence, which, despite notable successes and failures over decades still lacks formalization. Once a formalization is established, there is a potential for knowledge to become more 'vertical' (cumulative) than 'horizontal' (new, possibly different approaches).

Theoretical computer science may well be a suitable model. Once a formal definition of a Turing machine was accepted, theorems began to emerge to clarify what their capabilities and incapabilities were. Even unprovable theses, like Church's Thesis, could at least be asserted with clarity and become vital to the field.

Most neural networks, even biological ones, exhibit a layered or clustered structure. This publication shows that for universality all neural networks can be regarded as clustered. This clustered nature is therefore chosen as the basis for a formal neural network framework.

2 Related work

The efforts of the committee on neural network standardization of the IEEE Neural Network Society (see [Eberhart-90]) is the only other[1] significant neural network standardization attempt. In addition to this, a comprehensive and periodically updated neural network handbook in preparation ([Fiesler-95]) is foreseen to devote space to neural network standardization issues on a regular basis.

Some of the currently available introductory books on neural networks contain a chapter of foundations, such as [Rumelhart-86], [Wasserman-89], [Dayhoff-90], [Hecht-Nielsen-90], and [Simpson-90], but these can not be considered neural network formalizations per se.

There have been a few publications which included a (formal) description of a neural network, although most of them address only a limited class of neural networks. Jiawei Hong [Hong-88] describes in his paper a "connectionist computational machine", which can be regarded as a formal neural network definition. Clingman and Friesen describe in their group theoretical paper [Clingman-89] "learning automata" and address neural networks as an application of their theory. They only consider three layer neural networks and assume back-propagation as the learning rule. Valiant [Valiant-88] defines a "neural tabula rasa", which is roughly a sparse asymmetric neural network with real-valued weights.

[1] A short preliminary version of this paper has been published as [Fiesler-92].

All these attempts focus on the training of the neural network, while the underlying topological description is limited in scope. For example, none of the approaches addresses higher order (see section 4.5), ontogenic (see section 6), or slabbed (see section 4.2) neural networks.

3 Notation

An important step towards standardization is the use of a convenient and consistent notation. For a notation to be convenient, it should have symmetry and it should be mnemonic. *Symmetry* means that similar entities should have similar notations, and *mnemonic* means that the chosen symbol or variable should reflect its meaning. For example the first letter(s) of a word that describes an entity should be chosen as the parameter that represents that entity. For instance: N, representing the number of neurons in the network.

An other important issue is indexation. For example, the number of neurons in a specific layer should be denoted as an indexed symbol, where its index indicates the layer number. N_2 could therefore indicate the number of neurons in the second layer. If a different symbol is chosen for each layer, as seen in most neural network publications, besides being less clear, one soon runs out of symbols (for multi-layer neural networks). In general, when using indices, one should avoid the use of 'superscript' indices as much as possible because of the confusion with exponentiation.

As an example, observe the notation of one of the most important concepts in neural computing: that of an (inter)connection strength or weight. In order to specify a specific weight (or connection) in a first order neural network, four indices are needed: the index numbers of the layer where the connection originates (l) and where it terminates (m), plus both the index numbers of the neurons within these layers (i and j). Using the mnemonic W to represent a weight, leads to the notation $W_{l_i m_j}$. In case the network contains only interlayer connections (see section 4.3), the notation could be abbreviated to $W_{l,ij}$, with the assumption that either l indicates the layer where the connection originates, or where it terminates. The indicated assumption is a source of confusion, which is worse for asymmetric neural networks (see section 4.4). For single layer neural networks, the notation can be further reduced to W_{ij}.

Appendix A shows the proposed symmetric and mnemonic notation.

4 Neural network topology

The *topology* of a neural network is a combination of its framework (the neurons) and its interconnection scheme.

4.1 Frame(work)

The *frame* or *framework* of a neural network is defined by:

1. the number of clusters (layers or slabs)
2. the number of neurons per cluster.

The framework can be described by the set $\{N_1, N_2, \cdots, N_L\}$, where L is the total number of clusters in the neural network[2], and the size of the individual clusters, which is the number of neurons per cluster, is denoted by N_l, where l is the cluster number ($1 \leq l \leq L$). If the clusters can be linearly ordered they are called *layers* and the network is called a *layered neural network*. Otherwise, the clusters are called *slabs* and the network is called a *slabbed neural network* (see section 4.2). The *interconnection scheme* of a neural network is determined by the following four properties[3]:

1. the types of connections used (interlayer, intralayer, supralayer) (section 4.3),
2. whether the connections are symmetric or asymmetric (section 4.4),
3. the order of the connections (section 4.5), and
4. the *connectivity*, that describes which neurons are connected together.

4.2 Layers versus slabs

In most neural network architectures the neurons are clustered into layers, or more generally into slabs. In case a neural network architecture has no explicit clustering of neurons, it can be seen as a neural network consisting of one cluster. A *slab* is a collection of neurons that have a similar function, or hierarchical level. Layered neural networks are a more specific case; in a *layered neural network*, the clusters are ordered and can be numbered: the input layer[4] ordinal number 1, the first hidden layer, number 2, the second hidden layer, number 3, and so on, until the output layer, which has ordinal number L. N is the total number of neurons in the network, and therefore

$$N = \sum_{l=1}^{L} N_l. \tag{1}$$

The neurons within a layer (or slab) are not necessarily ordered, they are interchangeable by default. They are often numbered for reference purposes, though.

[2]Since the majority of all neural networks are layered, the symbol L is chosen to represent either the number of layers or the number of clusters.

[3]Hybrid architectures which incorporate different interconnection schemes within the same network, should be treated as a collection of subnetworks, each of which has a uniform interconnection scheme.

[4]Note that the input and output layers are counted as layers (see section 4.4).

4.3 Types of connections

In layered neural networks one can discriminate three types of connections:

1. Definition : An *interlayer connection* is a connection between neurons in adjacent layers[5].

   Note that a multi-layered neural network has at least one interlayer connection between two adjacent layers.

   The most commonly used neural networks, the (error) backpropagation neural networks [Werbos-74] [Rumelhart-86] are usually fully *interlayer connected*.

2. Definition : An *intralayer connection* is a connection between neurons of the same layer.

   Examples of neural networks that use intralayer connections are Brain-State-in-a-Box [Anderson-77] and the Adaptive Resonance Theory networks [Carpenter-86].

   A sub-class of intralayer connections are self-connections:

   Definition : A *self-connection* is a connection which originates and terminates at the same neuron. It provides direct feedback to the neuron itself (*self-inhibition* or *self-excitation*).

3. Definition : A *supralayer connection* is a connection between neurons that are neither in adjacent layers, nor in the same layer. That is, they 'skip' at least one layer.

   Multi-layer *recurrent neural networks* (see section 4.4) often have supralayer connections.

A neural network can have all possible connections:
Definition : A *plenary neural network* is a neural network which has all possible interlayer, intralayer, and supralayer connections. In other words it is a truly fully connected neural network.

For slabbed neural networks, some of these definitions have to be adapted. Due to the lack of order of slabs, only two kinds of connections can be discerned in slabbed neural networks: an *interslab connection* connects neurons from different slabs, and an *intraslab connections* connects neurons within the same slab with each other. A slabbed neural network with all possible interslab and intraslab connections is a *plenary (slabbed) neural network*.

Multidirectional Associative Memory neural networks [Hagiwara-90] are an example of fully interslab connected neural networks.

[5] *Adjacent* layers are layers that differ by one in ordinal number.

4.4 Symmetry versus asymmetry

A connection can be either symmetric or asymmetric:
Definition : A *symmetric* or *bidirectional connection* is a connection that has the same weight when used in either direction.
Definition : An *asymmetric* or *unidirectional connection* is a connection that can only be used for traversal in one direction, like the connections in *feed-forward neural networks*.

Neural network architectures with asymmetric connections can have two weights associated with each pair of connected neurons; one for each direction of traversal.

Note that the potential for a layered neural network to have bidirectional information propagation in general, or intralayer connections in specific, makes the input layer (layer one) a processing layer. This counters arguments of not counting the input layer as a layer.
Definition : A *symmetric neural network* is a neural network that contains solely symmetric connections.

A symmetric neural network has one or more symmetric weight matrices.
Definition : An *asymmetric neural network* is a neural network with one or more asymmetric connections.
Definition : A *recurrent neural network* is either an asymmetric neural network that contains at least one circuit[6] or a symmetric neural network.

For example, a neural network with self-connections is recurrent, since a self-connection is a circuit in itself.

4.5 Order

In order to use the available information entering a neuron more extensively, high(er) order connections [Giles-87] or *Sigma-Pi* connections [Williams-83] can be used;
Definition : A *high(er) order connection* is a connection that combines inputs from several neurons by means of a *splicing function*, which is usually multiplication. The number of inputs determines the order of the connection.
Definition : A *neural network of order* Ω is a neural network whose highest order connection has order Ω. A neural network with at least one high order connection is called a *high(er) order neural network*.

Most (traditional) neural networks are first order neural networks. A special kind of neural networks with splicing functions based on exponentiation[7] is described in [Durbin-89]. Note that splicing functions can also be applied to first order connections (see [Pao-88] [Pao-89]).

[6] *Circuit* as defined in graph theory.
[7] The order of these kind of connections could be calculated as the sum of the exponents.

5 Topological taxonomy

Many connectionist textbooks have adopted some basic classification of neural networks; usually learning rule based. The problem with these classifications is that neural networks can often not be uniquely classified due to their multi-faceted nature. In order to construct a formal neural network taxonomy it should be based on unique and precisely defined characteristics. Such a taxonomy, based on neural network topologies and supported by the definitions of section 4, is presented here in the form of a determination consisting of IF-THEN-ELSE clauses. Starting with number one, these clauses should be invoked in subsequent order, unless otherwise indicated by "GO TO" statements.

Neural network topology determination

1. IF the neural network is not clustered
   THEN GO TO 5.

2. IF the neural network is layered
   THEN classify as **layered**
   ELSE classify as **slabbed** AND GO TO 4.

3. IF the neural network has at least one supralayer connection
   THEN classify as **with supralayer connection(s)**.

4. IF the neural network has at least one intralayer connection
   THEN classify as **with intralayer connection(s)**
   ELSE GO TO 6.

5. IF the neural network has at least one self-connection
   THEN classify as **with self-connection(s)**
   ELSE classify as **without self-connection(s)**.

6. IF the neural network has at least one asymmetric connection
   THEN classify as **asymmetric**
   ELSE classify as **symmetric** and **recurrent**, AND GO TO 9.

7. IF the neural network has at least one loop (circuit)
   THEN classify as **recurrent**.

8. IF the highest order among all the connections is Ω
   THEN classify as **order** Ω.

9. END of determination.

A neural network topology can be classified by following the above steps and collecting the characteristics printed in bold face.

The taxonomy can be further extended by discriminating between fully and non-fully connectedness with respect to the network's intra-, inter-, and supra-layer connections.

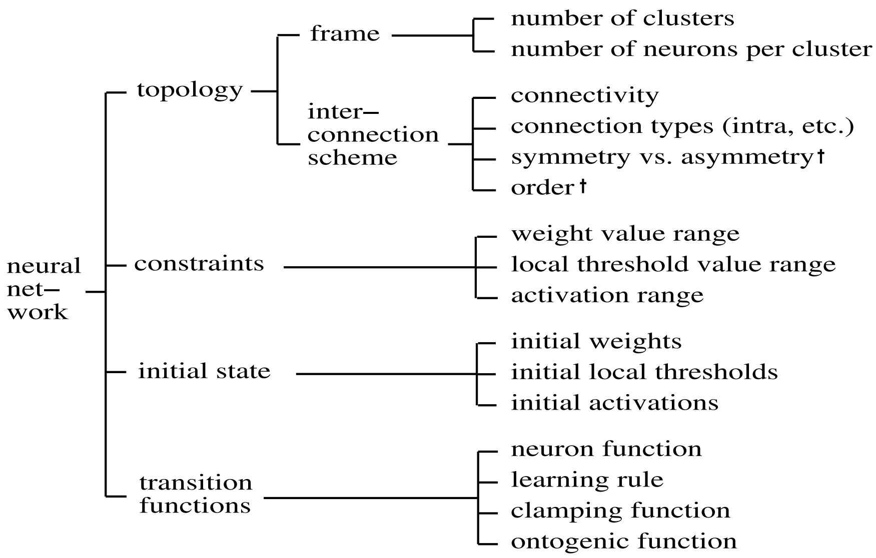

Figure 1: Hierarchical specification of a neural network.
† These aspects can be defined globally (at network level) or locally (at connection level).

6 A hierarchical description of a neural network

Based on the definitions of section 4, a hierarchical neural network description can be constructed. Figure 1 shows that the neural network specification can be split into a topology, a set of constraints, an initial state, and a set of transition functions. The topology of the neural network is defined in section 4. The constraints define the value ranges for the weights, the local thresholds (or biases or offsets), and the activation values (or activities), which can be regarded as the 'output' values of the neurons. Example constraints are the set of real values a continuous weight can assume, and a limited set of integers that discrete weights may assume (see [Fiesler-94]).

The *dynamics* of a neural network can be fully described by the initial state of the network, consisting of the set of initial values for the weights and local thresholds, plus the input (and target) patterns, which typically determine the (input neuron) activations, plus the transition functions which can be used to determine the successive states of the neural network, respecting the given constraints.

Known transition functions are:

1. *neuron functions*[8] (or *transfer functions*), which specify the output of a neuron, given its inputs (this includes the non-linearity),

[8] A neural network can have different neuron functions for different clusters.

2. *learning rules*[9] (or *learning laws*), which define how weights (and offsets) will be updated,

3. *clamping functions*, which determine if and when certain neurons will be insusceptible to incoming information, that is, they retain their present activation value independent of external stimuli, and

4. *ontogenic functions*, which specify changes in the neural network topology.

The first two transition functions are found in almost all artificial neural networks, the latter two in specific models only.

7 A formal neural network definition

This section describes a universal formal definition of a neural network. Like all formal definitions, it involves symbols but does not restrict the set of 'objects' that may be symbolized or the mechanisms whereby the formally defined entities may be realized. It is broad enough to encompass both simple biological neural networks, and essentially all artificial neural networks.

The definition consists of a collection of lines, each consisting of a descriptor, its symbolic notation, and its definition, sometimes followed by remarks. The definition part can contain new symbols (*non-terminals*) which will be defined on later lines (in depth first order).

A neural network ($\mathcal{NN}$) can be described as a four tuple: its topology, its constraints, its initial state, and its transition functions. The topology is a tuple in itself: framework plus interconnection structure (see section 4). A framework is defined by its clusters, which are sets of neurons. There are L clusters and N neurons in total. The interconnection structure consists of a set of relations between sets of source neurons ($\omega_{l,i}$) and terminal neurons ($n_{m,j}$). Source neurons are a subset of neurons from a cluster, say l, which supply information for their terminal neuron ($n_{m,j}$). The number of source neurons is equal to the order of the connection, and the order of the neural network (Ω) is defined as the maximum cardinal number of all the sets of source neurons. The constraints, initial state, and transition functions are discussed in section 6.

In the following definition, $\parallel S \parallel$ denotes the cardinal number of set S, which is the number of elements in S; an arrow ($\rightarrow$) denotes a mathematical relation; $\mathcal{Z}$ represents the set of all integers and $\mathcal{R}$ represents the set of all real numbers. The ellipsis ($\cdots$) in 'the constraints' section indicates that other algebraic number sets can also be used. $n_{l,i}$ denotes the ith neuron of the lth cluster, and $1 \leq l, m \leq L$, and $1 \leq i \leq N_l$, and $1 \leq j \leq N_m$. Note that for layered neural networks:

$$\begin{aligned}
|l-m| = 0 &\;\Leftrightarrow\; \text{intralayer connection}\\
|l-m| = 1 &\;\Leftrightarrow\; \text{interlayer connection}\\
|l-m| > 1 &\;\Leftrightarrow\; \text{supralayer connection}
\end{aligned}$$

[9] *Hybrid neural networks* can have different learning rules for different clusters.

The following scheme shows the mathematical definition of a neural network.

Neural Network	$\mathcal{NN}$	$= (T, C, S(0), \Phi)$	
The topology:			
Topology	T	$= (F, I)$	
Frame(work)	F	$= \{c_1, \ldots, c_L\}$	$\parallel F \parallel = L$
Cluster	c_l	$= \{n_{l,i}\}$	$\parallel c_l \parallel = N_l;$ $\sum_{l=1}^{L} \parallel c_l \parallel = \sum_{l=1}^{L} N_l = N$
Interconnection structure	I	$= \{\omega_{l,i} \rightarrow n_{m,j}\}$	$\parallel I \parallel = W$
Source neurons	$\omega_{l,i}$	$= \{n_{l,i}\} \subseteq c_l$	$\max_{l,i}(\parallel \omega_{l,i} \parallel) = \Omega$
The constraints:			
Constraints	C	$= \{C_W, C_\Theta, C_A\};$ $\{C_\gamma \mid (C_\gamma \subseteq \mathcal{Z}) \vee (C_\gamma \subseteq \mathcal{R}) \vee \cdots\},$ $\gamma \in \{W, \Theta, A\}$	
The initial state:			
Initial state	$S(0)$	$= \{W(0), \Theta(0), A(0)\}$	
Initial weights	$W(0)$	$= \{W_{\omega_{l,i} m_j}\} \in C_W$	$W(0) = I \rightarrow C_W$
Initial local thresholds	$\Theta(0)$	$= \{\theta_{l,i}\} \in C_\Theta$	$\Theta(0) = F \rightarrow C_\Theta$
Initial activity	$A(0)$	$= \{a_{1,i}\} \in C_A$	$A(0) = F \rightarrow C_A$
The transition functions:			
Transition functions	Φ	$= \{nf, lr, cf, of\};$	
	nf	= neuron function	
	lr	= learning rule	
	cf	= clamping function	
	of	= ontogenic function	

8 Summary

This publication presents a formalization and topological taxonomy of neural networks. The formalization consists of a uniform nomenclature and a mnemonic notation supplemented with both a hierarchical and a universal mathematical definition of a neural network. Using this formalization, a topology based neural network classification scheme is developed.

Acknowledgements

The author would like to thank Dr. Caulfield and Dr. Saxena for their comments and suggestions concerning a preliminary version of this document.

Appendix A The notation

Lower-case Roman letters i, j, l, and m are used as index variables; subsequent letters are used for subsequent entities of the same class.

Symbol	**Explanation**
$a_{l,i}$	activity (or activation value) of neuron i in layer l
L	number of layers (clusters) in neural network
N	total number of neurons in neural network
N_1	number of neurons in input layer (layer 1)
N_l	number of neurons in layer l
N_L	number of neurons in output layer (layer L)
$\omega_{l,i}$	set of source neurons; contains all the neurons a specific connection originates from
Ω	order of neural network[10]
$\theta_{l,i}$	local threshold (or bias or offset) of neuron i in layer l
W	total number of weights in neural network
$W_{l_i m_j}$	a weight between neuron i of layer l and neuron j of layer m
$W_{\omega_{l,i} m_j}$	a higher order weight between source neurons $\omega_{l,i}$ of layer l and neuron j of layer m

References

[Anderson-77] James A. Anderson, Jack W. Silverstein, Stephen A. Ritz, and Randall S. Jones, **Distinctive Features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model**, *Psychological Review*, volume 84, pages 413–451, 1977.

[Carpenter-86] Gail A. Carpenter and Stephen Grossberg, **Adaptive Resonance Theory: Stable Self-organization of Neural Recognition Codes in Response to Arbitrary Lists of Input Patterns**, *Proceedings of the Eight Annual Conference of the Cognitive Science Society*, pages 45–62, Lawrence Erlbaum Associates, Hillsdale, New Jersey, U.S.A., 1986.

[Clingman-89] William H. Clingman and Donald K. Friesen, **Algebraic Analysis of Neural Net Learning**, *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks (IJCNN-89)*, volume II, page 624 (abstract only),

[10] For correspondence restricted to the ASCII character set, a capital 'O' could for example be used instead of the letter Ω. In that case, one should be aware of the resemblance between the number '0' and the letter 'O'.

Washington, D.C., June 18–22, 1989, IEEE TAB neural network committee; SOS Printing, San Diego, California, 1989, IEEE Catalog number 89CH2765-6.

[Dayhoff-90] Judith Dayhoff, **Neural Network Architectures; An Introduction**, Van Nostrand Reinhold, New York, New York, 1990, ISBN: 0-442-20744-1.

[Durbin-89] R. Durbin and David E. Rumelhart, **Product Units: A Computationally Powerful and Biologically Plausible Extension to Backpropagation Networks**, *Neural Computation*, volume 1, pages 133–142, MIT Press, Cambridge, Massachusetts, 1989.

[Eberhart-90] Russell C. Eberhart, **Standardization of Neural Network Terminology**, *IEEE Transactions on Neural Networks*, volume 1, number 2, pages 244–245, June 1990.

[Fiesler-92] E. Fiesler and H. J. Caulfield, **Layer Based Neural Network Formalization**, *Artificial Neural Networks II*, volume 1, pages 329–332, editors: I. Alexander and J. G. Taylor, North-Holland / Elsevier Science Publishers, Amsterdam, The Netherlands, 1992, ISBN: 0-444-89488-8.

[Fiesler-94] E. Fiesler, A. Choudry, and H.J. Caulfield, **A Universal Weight Discretization Method for Backpropagation Neural Networks**, accepted by: *IEEE Transactions on Systems, Man, and Cybernetics*.

[Fiesler-95] E. Fiesler and R. Beale (editors), **Handbook of Neural Computation**, Oxford University Press and Institute of Physics Publishing.

[Giles-87] C. Lee Giles and Tom Maxwell, **Learning, Invariance, and Generalization in High-Order Neural Networks**, *Applied Optics*, volume 26, number 23, pages 4972–4978, December 1, 1987.

[Hagiwara-90] Masafumi Hagiwara, **Multidirectional Associative Memory**, *Theory Track, Neural and Cognitive Sciences Track of the Proceedings of the International Joint Conference on Neural Networks (IJCNN-90-WASH-DC)*, volume I, pages 3–6, Maureen Caudill (editor), Washington, District of Columbia, January 15–19, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990, ISBN: 0-8058-0775-6.

[Hecht-Nielsen-90] Robert Hecht-Nielsen, **Neurocomputing**, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990, ISBN: 0-201-09355-3.

[Hong-88] Jiawei Hong, **On Connectionist Models**, *Communications on Pure and Applied Mathematics*, volume 41, pages 1039–1050, 1988.

[Pao-88] Yoh-Han Pao, CAISR Technical Report, number 88-105, Center for Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland, Ohio, 1988.

[Pao-89] Yoh-Han Pao, **Adaptive Pattern Recognition and Neural Networks**, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1989, ISBN: 0-201-12584-6.

[Rumelhart-86] David E. Rumelhart, James L. McClelland, and the PDP Research Group, **Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations**, The MIT Press, Cambridge, Massachusetts, 1986, ('A Bradford Book'), ISBN: 0-262-18120-7.

[Simpson-90] Patrick K. Simpson, **Artificial Neural Systems; Foundations, Paradigms, Applications, and Implementations**, series: "Neural networks: research and applications", Pergamon Press, Elmsford, New York, 1990, ISBN: 0-08-037895-1.

[Valiant-88] L. G. Valiant, **Functionality in Neural Nets**, *Proceedings of the American Association for Artificial Intelligence (AAAI)-88*, volume 2, pages 629–634, Seventh National Conference on Artificial Intelligence, St. Paul, Minnesota, August 21–26, 1988, Morgan Kaufmann publishers, San Mateo, California, 1988, ISBN: 0-929280-00-8.

[Wasserman-89] Philip D. Wasserman, **Neural Computing; Theory and Practice**, Van Nostrand Reinhold, New York, New York, 1989, ISBN: 0-442-20743-3.

[Werbos-74] Paul Werbos, **Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences**, Ph.D. dissertation, Harvard University, 1974.

[Williams-83] Ronald J. Williams, **Unit Activation Rules for Cognitive Network Models**, ICS Technical Report number 8303, Institute for Cognitive Science, University of California, San Diego (UCSD), La Jolla, California, 1983.