Neural Network Formalization

E. Fiesler

Institut Dalle Molle d'Intelligence Artificielle Perceptive (IDIAP)
Case postale 609, CH-1920 Martigny, Switzerland
Electronic mail: EFiesler@IDIAP.CH (Internet)

Abstract

In order to assist the field of neural networks in its 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.

Most neural networks, even biological ones, exhibit a layered structure. This publication shows that all neural networks can be represented as layered structures. This layeredness is therefore chosen as the basis for a formal neural network framework. 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 statics

[0]An improved and expanded version of this technical report has been published: E. Fiesler. **Neural Network Classification and Formalization**. *Computer Standards & Interfaces*, special issue on Neural Network Standards, John Fulcher (editor), volume 16, number 3, pages 231–239. North-Holland / Elsevier Science Publishers, Amsterdam, The Netherlands, June 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 varigated terminology and synonymic nomenclature developed. Coinciding, 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. Leading neural network proponents, like Stephen Grossberg and Bart Kosko, have argued that neural networks differ from artificial intelligence by having formal theorems like the Cohen-Grossberg theorem and ABAM theorem [Kosko-90], rather than simply *ad hoc* methods and procedures. Once a formalization is established, an increase can be expected to be seen in the 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 structure. This publication shows that for universality all neural networks can be regarded as layered. This layeredness is therefore chosen as the basis for a formal neural network framework.

2 Prior work

The efforts of the committee on neural network standardization of the IEEE Neural Network Society (cf. [Eberhart-90]) is the only other significant neural network standardization attempt. (Some of the 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. Usually they 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 (NTR)", which is roughly a sparse asymmetric neural network with real-valued weights.

All these attempts focus on the dynamics (training) of the neural network, while the underlying static description is limited in scope. For example, none of the approaches addresses higher order (see section 4.5), ontogenic (see section 5), or slabbed (see section 4.2) neural networks.

3 Notation

A first step towards standardization should be the use of a convenient and consistent notation. For a notation to be convenient, it should have symmetry and it should be mnemonical. *Symmetry* means that similar entities should have similar notations, and *mnemonical* 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.

An 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 a 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 $W_{l_i m_j}$. If the network contains only interlayer connections (see section 4.3), the notation can 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 asymetric 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 (symmetrical and mnemonical) notation.

4 Neural network statics

Neural network statics complements the field of neural network dynamics. It deals with network parameters which remain constant during the training and recall phases. In this publication, the network architecture is assumed to be part of the neural network statics. Neural network architectures can be classified by their topology.

4.1 Neural network topology

The *topology* of a neural network is a combination of its frame and its interconnection scheme.

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

1. the number of clusters (i.e. layers or slabs)

2. the number of neurons per cluster

The framework can be described by the (ordered) set $\{N_1, N_2, \cdots, N_L\}$, where L is the total number of clusters in the neural network[1], 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 are 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[2]:

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*, i.e. which neurons are connected.

These topological properties can serve as a taxonomy for neural networks.

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, they can be seen as neural networks consisting of one

[1] Since the majority of all neural networks are layered, L is chosen to represent the number of layers (clusters).

[2] 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.

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[3] 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. They are often numbered for reference purposes, though.

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 of the neural network.

   The most commonly used neural networks, such as the multi-layer perceptron [Rosenblatt-58] and the standard back(ward error) propagation neural network [Werbos-74] [Rumelhart-86] are usually fully *interlayer connected.*

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

   Brain-State-in-a-Box (BSB) [Anderson-77] and Adaptive Resonance Theory (ART) [Carpenter-86] are just two of the better known neural networks that use intralayer connections. 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 of the neural network, i.e. they 'skip' at least one layer.

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

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

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 (MAM) neural networks [Hagiwara-90] are an example of fully interslab connected neural networks.

4.4 Symmetry versus asymmetry

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

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.

Neural network architectures with asymmetric connections can have two weights associated with each pair of connected neurons: one is used in the forward propagation, the other in the backward propagation. (If a neural network with asymmetric connections uses only unidirectional propagation, the interconnection topology of the neural network is equivalent to one with symmetric connections.)
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* has one or more asymmetric connections.
Definition : A *recurrent neural network* is either a symmetric neural network which contains at least one cycle[4] or an asymmetric neural network that contains at least one circuit.

For example, a neural network with self-connections is always recurrent, due to the implicit asymmetry of self-connections.

[4] *Cycle* and *circuit* as defined in graph theory.

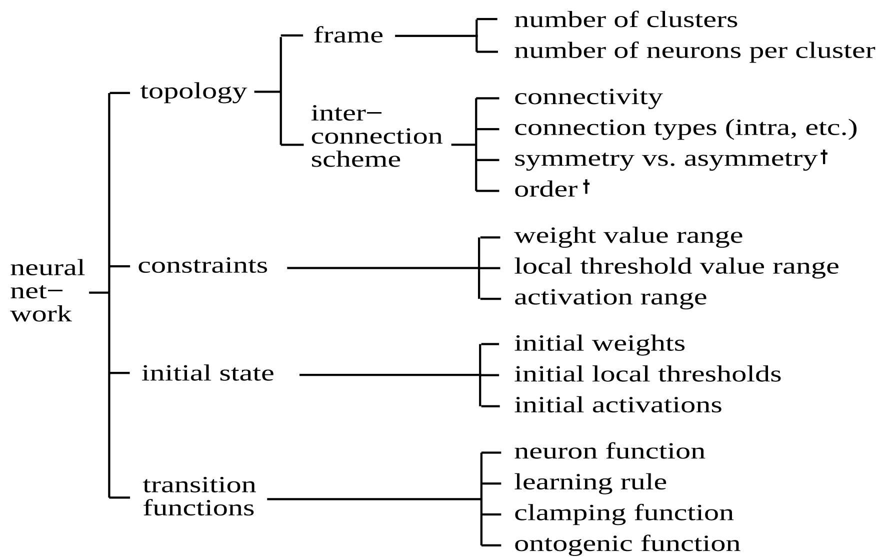

Figure 1: A hierarchical specification of a neural network.
† These aspects can be defined globally (at network level) or locally (at connection level).

4.5 Order

In order to use the available information entering a neuron more extensively, so called high(er) order connections [Giles-87] can be defined:
Definition : A *high(er) order connection* is a connection that combines inputs from several neurons; usually by 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 order two or higher is called a *high(er) order neural network*.

Most (traditional) neural networks are first order neural networks.

5 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 static part and a dynamic part. The neural network statics (cf. introduction of section 4) comprises both the topology and the constraints. The constraints define the value ranges for the weights, the local thresholds (or offsets), and the activation values (or activities), which can be seen 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 (cf. [Fiesler-92]). The topology of the neural network is defined in section 4.

The dynamics of a neural network can be fully described by an initial state, consisting of a set of initial values for the weights, local thresholds, and (input neuron) activations, plus transition functions which can be used to determine the successive states of the neural network. Known transition functions are:

1. *neuron functions*[5] (or *transfer functions*), which specify the output of a neuron, given its inputs (this includes the non-linearity),

2. *learning rules* (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, i.e. they retain their present activation value, and

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

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

6 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 (cf. section 4.1). A frame(work) 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

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

defined as the maximum cardinal number of all the sets of source neurons. The constraints and dynamics are discussed in section 5.

In the following definition, $\| S \|$ denotes the cardinal number of set S, i.e. 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 ($\cdots$) near the end of 'the statics' indicates that any other number set 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:

$|l - m| = 0$: intralayer connection
$|l - m| = 1$: interlayer connection
$|l - m| > 1$: supralayer connection

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

Neural Network	$\mathcal{NN}$	$= (T, C, s(0), \Phi)$	
The statics:			
Topology	T	$= (F, I)$	
Frame(work)	F	$= \{c_1, \ldots, c_L\}$	$\| F \| = L$
Cluster	c_l	$= \{n_{l,i}\}$	$\| c_l \| = N_l$; $\sum_{l=1}^{L} \| c_l \| = \sum_{l=1}^{L} N_l = N$
Interconnection structure	I	$= \{\omega_{l,i} \rightarrow n_{m,j}\}$	$\| I \| = W$
Source neurons	$\omega_{l,i}$	$= \{n_{l,i}\} \subseteq c_l$	$\max_{l,i}(\| \omega_{l,i} \|) = \Omega$
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 dynamics:			
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	$A(0)$	$= \{a_{l,i}\} \in C_A$	$A(0) = F \rightarrow C_A$
Transition functions	Φ	$= \{nf, lr, cf, of\}$;	
	nf	= neuron function	
	lr	= learning rule	
	cf	= clamping function	
	of	= ontogenic function	

7 Summary

This publication presents a neural network formalization based on the concept of layeredness. It consists of a mnemonic notation, a uniform nomenclature, and a topological taxonomy, supplemented with both a hierarchical and a universal mathematical definition of a neural network.

Appendix A The notation

$a_{l,i}$	activity (or activation value) of neuron i in layer l
L	number of layers (clusters) in neural network
N	total number of neurons (clusters) 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'; i.e. all the neurons a specific connection originates from
Ω	order of neural network
$\theta_{l,i}$	local threshold (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

The lowercase roman letters i, j, l, and m are used as index variables; subsequent letters are used for subsequent entities of the same class.

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), 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.

[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, A. Choudry, and H.J. Caulfield, **A universal weight discretization paradigm for backpropagation neural networks**, accepted by: *IEEE Transactions on Systems, Man, and Cybernetics.*

[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, editor: Maureen Caudill, at: Washington, District of Columbia, January 15-19, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990, ISBN: 0-8058-0775-6 (volume 1) or 0-8058-0754-3 (set of two volumes).

[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.

[Kosko-90] Bart Kosko, Personal communication with the author.

[Rosenblatt-58] Frank Rosenblatt, **The perceptron: a probabilistic model for information storage and organization in the brain**, *Psychological Review*, volume 65, pages 386-408, 1958.

[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 (volume 1) or 0-262-18123-1 (set of two volumes).

[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.