Comparative Bibliography of Ontogenic Neural Networks

E. Fiesler

IDIAP, Case postale 609, CH-1920 Martigny, Switzerland

One of the most powerful aspects of neural networks is their ability to adapt to problems by changing their interconnection strengths according to a predetermined *learning rule*. On the other hand, one of the main drawbacks of neural networks is the lack of knowledge for determining the topology of the network, that is, the number of layers and number of neurons per layer. A special class of neural networks tries to overcome this problem by letting the network also automatically adapt its topology to the problem. These are the so called ontogenic neural networks. Other potential advantages of ontogenic neural networks are improved generalization, implementation optimization (size and/or execution speed), and the avoidance of local minima.

Possible topology modifications are the addition and/or removal of layers, neurons, or connections. Note that modification of the number of layers usually implies modification of the number of neurons, and that modification of the number of neurons usually implies modification of the number of connections. This publication provides an extensive comparison of these ontogenic neural networks in a compact tabular form.

In this table, the following notations are used. "*L*" is the number of layers (including the input and output layer), $l+x$ means that the number of layers is initially l and can grow. The arrow-columns indicate whether the method increases ($\nearrow$) or decreases ($\searrow$) the network size by manipulating layers (L), neurons (N), or weights (W). (The symbol "N" is also used in the case of an initial two-layer network growing to become a three-layer network with the addition of the first neuron.) "*Lo.*" indicates whether the method is local, which means that the method is exclusively based on information available at the neuron in question, which includes information about the connections to its neighbors. "*I.*" denotes the initial conditions of the method which are encoded by a number, indicating the number of layers in the initial topology (0 for an empty topology), a first letter, indicating the initial connectivity ("F" for fully interlayer connected, "T" for a tree structure, "O" for other topologies), and a second letter, indicating the values of the initial weights ("R" when they are initialized with random values, "T" if the ontogenic method starts with an already trained network). And lastly, "*Pr.*" lists whether a convergence proof exists (M=mathematical proof, I=informal proof). In general, an "*X*" indicates that a condition is fulfilled, while a "-" indicates it is not. A "?" denotes that the information is not available in the quoted reference.

In order to sort the various kinds of ontogenic neural networks, they are classified firstly by learning rule (based on Perceptron, feature map, or other) and secondly by topology (layered or tree based), and thirdly by growing and/or shrinking.

The main goal for most **Perceptron based growing methods** is the determination of the (optimal) topology of a neural network, followed by the goal to avoid getting trapped in local minima, where new units are added in the hope that a higher dimensional error surface eliminates the local minima in question. It is interesting to note that all the tree based methods are growing methods.

Perceptron based pruning methods are usually targeted towards improving generalization, since a small network usually has a better generalization capability.

Most **Perceptron based methods that both add and delete units** are a mere combination of growing and pruning methods. Firstly, they increase the network size

[0]This document will be published in: *Proceedings of the International Conference on Artificial Neural Networks (ICANN '94).*

until it is large enough to handle the problem (or to eliminate possible local minima). This often yields an oversized topology. Next, redundant units are removed.

Reference	*Model*	*Name*	*L*	$\nearrow$	$\searrow$	*Lo.*	*Init.*	*Pr.*
		Perceptron based methods						
[Ash-89.2]	BP[1]	Dyn. Node Creation	3	N	-	-	3FR	-
[Bellido-91]	BP	-	3	N	-	X	2FR	-
[Fahlman-90.2]	MLP[2]	Cascade Correlation	$2{+}x$	L		X	2FR	-
[Fahlman-91.2]	MLP	Recurrent CasCor.	$2{+}x$	L	-	-	2OR	-
[Hanson-90]	SDR[3]	Meiosis networks	3	N	-	X	2FR	-
[Mezard-89]	MLP	Tiling	$2{+}x$	L	-	?	2FR	M
[Nadal-89]	MLP	-	$2{+}x$	L	-	?	xOR	M
[Rujan-89.2]	MLP	regular partitioning	3	N	-	-	2FR	M
[Sirat-90]	MLP	-	3	N	-	X	1	I
[Zollner-92]	MLP	-	3	N	-	-	2FR	I
[Thodberg-91]	BP	Ockham's Razor	3	-	W	-	3FT	-
[LeCun-90]	MLP	Opt. Brain Damage	3	-	W	-	3FR	-
[Hassibi-93]	MLP	Opt. Brain Surgeon	3	-	W	-	3FR	-
[Hagiwara-90.2]	BP	-	3	N	W	-	3FR	-
[Hirose-91]	BP	-	3	N	N	-	?	-
[Honavar-88] [Honavar-89.2]	'cones' BP	units recruitment	*L*	N[4]	N W	X	x O	-
[Deffuant-90]	BP	NEURAL[5]	tree	N	-	X	0	M
[Golea-90]	BP	decision trees	tree	N	-	X	2	M
[Frean-90]	BP	Upstart algorithm	tree	N	-	?	2OR	I
		Self-Organizing Feature Map based methods						
[Wen-92.1]	comb.[6]	SSGNN[7]	$1{+}x$	N	N	-	0	-
[Fritzke-91.1] [Fritzke-91.2] [Fritzke-91.3]	SOFM	-	2	N	N	X	1OR	-
		Other ontogenic neural networks						
[Lansner-87]	Hopfld.	-	2	N	-	X	2OR	-
[Kadir.-92.1] [Kadir.-92.2]	RBF[8]	-	3	N	-	-	2FR	-
[Bonnlander-93]	RBF	latticed networks	2	X	-	-	2OR	-
[Alpaydin-90.1]	-	Grow-and-Learn	3	N	N	-	1O	I
[Diederich-88]	-	Neuron Recruitment	4	N	-	-	-	-
[Reilly-82]	-	Category Learning	3	N	-	X	0	I
[Ivakhnenko-68]	GMDH	-	$1{+}x$	L	N	X	1	I
[Barron-75]	GMDH	McLaurin series	$1{+}x$	L	N	X	1	-
[Tenorio-89]	GMDH	Simulated Annealing	$1{+}x$	L	N	X	1	-
[Baum-91]	MLP	Query Learning	3	X	-	-	2FR	-
[Fiesler-92.2]	HONN[9]	Superceptron	2	W	-	-	0	-
[Ring-93.2]	HONN	-	$2{+}x$	L	-	-	2FR	-

[1] Backpropagation
[2] Multilayer Perceptron
[3] Stochastic Delta Rule
[4] transform = cluster of neurons
[5] NEural Units Recruitment ALgorithms
[6] combination of Self-Organizing Feature Map and Multilayer Perceptron
[7] Self-Generating Neural Networks
[8] Radial Basis Function
[9] High Order Neural Networks

The typically long training time for **Kohonen's Self-Organizing Feature Maps** can be shortened by starting with a small topology which grows according to some predefined criteria. Another reason for an ontogenic approach is to add neurons where the input space density is the greatest, in order to overcome the impossibility of dividing the input distribution into disjunct sub-structures. As is the case for Perceptron based neural networks, the generated network is often not optimal, and may be improved by removing neurons or otherwise reshaping the network.

The small capacity of **Hopfield Networks** can be improved with algebraic methods like the projection rule which reduce training set constraints.

The author acknowledges the support of J. Klotz, who participated in the creation of this document as well as its expanded and comprehensively annotated version, which is currently in preparation [Klotz-94].

References

[Alpaydin-90.1] E. Alpaydin. Grow-and-learn: An incremental method for category learning. In *Proc. INNC '90*, volume 2, pages 761–764, Dordrecht, The Netherlands, 1990. Kluwer.

[Ash-89.2] T. Ash. Dynamic node creation in backpropagation networks. *Connection Science*, 1(4):365–375, 1989.

[Barron-75] R. L. Barron. Learning networks improve computer-aided prediction and control. *Computer Design*, pages 65–70, August 1975.

[Baum-91] E. B. Baum and K. J. Lang. Constructing hidden units using examples and queries. In R. P. Lippmann et al., editor, *NIPS 3*. Morgan Kaufmann, 1991.

[Bellido-91] I. Bellido and G. Fernández. Backpropagation growing networks: Towards local minima elimination. In A. Prieto, editor, *Artificial Neural Networks; Proc. IWANN '91*, pages 130–135, Heidelberg, Germany, 1991. Springer.

[Bonnlander-93] B. Bonnlander and M. C. Mozer. Latticed rbf networks: An alternative to constructive methods. In *NIPS 5*, 1993.

[Deffuant-90] G. Deffuant. Neural units recruitment algorithm for generation of decision trees. In *Proc. IJCNN '90 - San Diego*, volume I, pages 637–642, Ann Arbor, MI, 1990. IEEE Neural Networks Council; Edward Brothers.

[Diederich-88] J. Diederich. Connectionist recruitment learning. In *Proc. of the 8th European Conference on A.I.*, pages 351–356, 1988.

[Fahlman-90.2] S. E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. Technical Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1990.

[Fahlman-91.2] S. E. Fahlman. The recurrent cascade-correlation architecture. Technical Report CMU-CS-91-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 17 1991.

[Fiesler-92.2] E. Fiesler. Partially connected ontogenic high order neural networks. Technical Report 92-02, IDIAP, Martigny, Switzerland, August 1992.

[Frean-90] M. Frean. The upstart algorithm: A method for constructing and training feedforward neural networks. *Neural Computation*, 2(2):198–209, 1990.

[Fritzke-91.1] B. Fritzke. Let it grow - self-organizing feature maps with problem dependent cell structure. In T. Kohonen et al., editor, *Artificial Neural Networks; (ICANN-91)*, volume 1, pages 403–408, Amsterdam, The Netherlands, 1991. North-Holland; Elsevier Science Publishing Company B.V.

[Fritzke-91.2] B. Fritzke. Unsupervised clustering with growing cell structures. In *Proc. IJCNN '91 Seattle*, volume II, pages 531–536, 1991.

[Fritzke-91.3] B. Fritzke and P. Wilke. Flexmap - a neural network with linear time and space complexity for the traveling salesman problem. In *Proc. IJCNN-91*, 1991.

[Golea-90] M. Golea and M. Marchand. A growth algorithm for neural network decision trees. *EuroPhysics Letters*, 12(3):205–210, 1990.

[Hagiwara-90.2] M. Hagiwara. Novel back propagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection. In *Proc. IJCNN '90 San Diego*, volume I, pages 625–630, Ann Arbor, MI, 1990. IEEE Neural Networks Council; Edward Brothers.

[Hanson-90] S. J. Hanson. Meiosis networks. In D. S. Touretzky, editor, *NIPS 2*, pages 533–541, San Mateo, CA, 1990. Morgan Kaufmann.

[Hassibi-93] B. Hassibi and D. G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *NIPS 5*, 1993.

[Hirose-91] Y. Hirose, K. Yamashita, and S. Hijaya. Back-propagation algorithm which varies the number of hidden units. *Neural Networks*, 4:61–66, 1991.

[Honavar-88] V. Honavar and L. Uhr. A network of neuron-like units that learns to perceive by generation as well as reweighting of its links. In D. Touretzky et al., editor, *Proc. of the 1988 Connectionist Models Summer School*, pages 472–484, San Mateo, California, 1988. Morgan Kaufmann.

[Honavar-89.2] V. Honavar and L. Uhr. Generative learning structures and processes for generalized connectionist networks. *Connection Science*, 1:139–159, 1989.

[Ivakhnenko-68] A. G. Ivakhnenko. The group method of data handling - a rival of stochastic approximation. *Soviet Automatic Control*, 1:43–55, 1968.

[Kadir.-92.1] V. Kadirkamanathan and M. Niranjan. Application of an architecturally dynamic network for speech pattern classification. In *Proc. of the Institute of Acoustics*, volume 14, 1992.

[Kadir.-92.2] V. Kadirkamanathan and M. Niranjan. A function estimation approach to sequential learning with neural networks. Technical Report CUED/F-INFENG/TR.111, Cambridge University Engineering Department, Cambridge, UK, September 13 1992.

[Klotz-94] J. Klotz and E. Fiesler. Ontogenic neural networks. *Expert Systems*. (to be submitted).

[Lansner-87] A. Lansner and Ö. Ekeberg. An associative network solving the "4-bit ADDER problem". In M. Caudill and C. Butler, editors, *Proc. ICNN*, volume II, pages 549–556, San Diego, CA, 1987. SOS Printing.

[LeCun-90] Y. Le Cun, J. S. Denker, and S. A. Solla. Optimal brain damage. In D. S. Touretzky, editor, *NIPS 2*, pages 598–605, San Mateo, CA, 1990. Morgan Kaufmann.

[Mezard-89] M. Mézard and J.-P. Nadal. Learning in feedforward layered networks: The tiling algorithm. *J. of Physics: A*, 22(12):2191–2203, 1989.

[Nadal-89] J.-P. Nadal. Study of a growth algorithm for a feedforward neural network. *Int. J. of Neural Systems*, 1(1):55–59, 1989.

[Reilly-82] D. L. Reilly, L. N. Cooper, and C. Elbaum. A neural model for category learning. *Biological Cybernetics*, 45:35–41, 1982.

[Ring-93.2] M. Ring. Sequence learning with incremental high-order neural networks. CSE technical Report AI 93-193, Dept. of Computer Sc., University of Texas at Austin, January 1993.

[Rujan-89.2] P. Ruján and M. Marchand. Learning by minimizing resources in neural networks. *Complex Systems*, 3:229–241, 1989.

[Sirat-90] J. A. Sirat and J. P. Nadal. Neural trees: A new tool for classification. *Network; Computation in Neural Systems*, 1:423–438, 1990.

[Tenorio-89] M. F. Tenorio and W.-T. Lee. Self organizing neural networks for the identification problem. In D. S. Touretzky, editor, *NIPS 1*, pages 57–64, San Mateo, CA, 1989. Morgan Kaufmann.

[Thodberg-91] H. H. Thodberg. Improving generalization of neural networks through pruning. *Int. J. of Neural Systems*, 1(4):317–326, 1991.

[Wen-92.1] W. X. Wen, H. Liu, and A. Jennings. Self-generating neural networks. In *Proc. IJCNN - Baltimore*, 1992.

[Zollner-92] R. Zollner, H. J. Schmitz, F. Wünch, and U. Krey. Fast generating algorithm for a general three-layer perceptron. *Neural Networks*, 5(5):771–777, September–October 1992.