Neural Networks and Evolutionary Computation. Part I: Hybrid Approaches in Artificial Intelligence

Gerhard Weiß

Abstract— **This paper series focusses on the intersection of neural networks and evolutionary computation. It is addressed to researchers from artificial intelligence as well as the neurosciences.**

Part I provides a comprehensive and compact overview of hybrid work done in artificial intelligence, and shows the state of the art of combining artificial neural networks and evolutionary algorithms.

Keywords— **Artificial neural network, evolutionary algorithm, evolutionary design, evolutionary training.**

I. Introduction

Artificial neural networks and evolutionary computation establish two major research and application areas in artificial intelligence. In analogy to biological neural networks, artificial neural networks (ANNs) are composed of simple processing elements that interact using weighted connections. ANNs are of particular interest because of their robustness, their parallelism, and their learning abilities; see e.g. [9, 26, 51] for introducing literature. Evolutionary computation is typically considered in the context of evolutionary algorithms (EAs). The most common forms of EAs are Rechenberg's evolution strategy (e.g., [49, 54]) and Holland's genetic algorithm (e.g., [11, 19, 30]). Although these forms differ with respect to several implementational details, conceptually they are nearly identical [29]. EAs establish a very general and powerful search, optimization and learning method that bases, in analogy to biological evolution, on the application of evolutionary operators like mutation, recombination and selection. Like no other computational method, EAs have been applied to a very broad range of problems [2].

In the recent years the idea of combining ANNs and EAs has received much attention [1, 32, 50, 58, 59, 69], and now there is a large body of literature on this subject. This paper overviews this literature and shows the state of the art of bringing ANNs and EAs together. The paper is organized as follows. Section II deals with the approaches to an evolutionary design of appropriate structures of ANNs. Section III summarizes the approaches to an evolutionary training of ANNs. Section IV provides a guide to further hybrid approaches in artificial intelligence that do not fall into the "mainstream categories" treated in the two preceding sections. Section V concludes the paper with some general remarks on the idea of synthesizing ANNs and EAs.

The author is with the Institut für Informatik (H2), Technische Universität München, D-80290 München, Germany.

II. Evolutionary Network Design

Extensive experimental data reported in the literature show that there is a strong connection between the structure (size and connectivity) and the function of ANNs. This connection is twofold: first, it concerns the performance of learning ("How appropriate is a network structure for learning a desired function?"); and second, it concerns the comprehensibility of representation ("How transparent or opaque is the learned function represented in the network structure?"). Unfortunately, apart from some vague statements (e.g., "networks being to large loose may loose their generalization ability" or "learning a function requires a larger network than representing it"), almost nothing is known about this structure–function connection, and there is no method for a priori specifying a network structure which is suitable with regard to learning performance or representational comprehensibility. Even after a decade of enormous progress in the field of ANNs, network design remains a critical point, and this causes a growing interest in the automated design of appropriate network structures. As it was pointed out in [40], there are several reasons for using EAs in automatically designing networks. In particular, both enumerative, random and gradient-descent search methods are limited in their application because the search space of all possible network structures is infinitely large, undifferentiable, deceptive and multi-modal. The following subsections describe approaches to and aspects of an evolutionary design of ANNs.

A. Design Criteria

There is an increasing number of approaches to an "evolution-based" design of neural network structures. These approaches can be grouped according to their design criteria into two broad classes. First, approaches whose design criterion is the **learning performance** (including aspects like speed, accurary, generalization ability); see e.g. [22, 24, 36, 38, 40, 47, 52, 68]. The general intention underlying these approaches is to find network structures which improve the learning performance of conventional neural learning procedures (e.g., backpropagation). Examples of application tasks used in these approaches are the standard XOR, two–dimensional XOR, TC discrimination, Boolean multiplexer and digit/face recognition. In almost all cases the evolved networks showed a significantly improved learning behavior compared with the initial networks. Interestingly, EAs often lead to appropriate structures which are quite different from those (e.g., layered feedforward or simple recurrent structures) typically used by human designers. On the other side, however, it is still unknown whether or not specific network structures evolve under different design criteria; this issue is addressed in [60].

Second, approaches whose design criterion is the **representational comprehensibility** of network structures. For instance, in [15, 16] the question is addressed how symbolic schemata might be implemented at the subsybmbolic, neural–like (connectionist) level. Thereby they took the view that networks learning by constructing opaque representations "may yield little to our understanding of human cognitive information processing" [16, p. 123] and that "in order to learn large symbolic structures of the type that people use, specific architectures will be required" [15, p. 8]. The primary intention for applying the EA technique was to demonstrate that there is a plausible evolutionary path along which network structures suitable for symbol processing can evolve. The experimental results indicated that the low–structured networks were less robust as regards mutational changes than the high–structured ones; particularly, these experiments showed an evolutionary tendency towards complex, hierarchically organized structures. Other hybrid approaches which can be viewed under the aspect of representational comprehensibility are described in e.g. [12, 13, 46].

B. Genotypic Representation of ANNs

The application of EAs requires an encoding of the network structures into specific representations or "genotypes" upon which the evolutionary operators mutation and recombination can act. The choice of the genotypic representation and the evolutionary operators is decisive to the efficiency of EAs. In particular, this choice affects the following **central aspects**:

- structural completeness ("Which structures of which size and connectivity are available?"),
- structural correctness ("Do all mutated and recombined genotypes specify correct structures?"),
- structural level of operator application ("At which network level – individual connections or whole sub-networks – do mutation and recombination operate?"), and
- structural sensibility to operator application ("To what degree do mutation and recombination influence the network structures?").

(Note that mutation and recombination are syntactic operators that they are applied to the genotypes without regard to the function or semantics of the phenotypes.)

Two different types of representational schemes have been propostedin the literature. First, the **low–level scheme** according to which the structure is specified more or less directly by the network connectivity (e.g., [40, 52]). And second, the **high–level scheme** according to which the structure is specified in a relatively abstract way either by network parameters (like the number of layers or units, the degree and the types of connectivity between and within the layers, the size of the units' receptive fields, and so forth, e.g. [15, 16, 24, 36, 38]) or by network growth rules (e.g. [20, 33, 41, 42, 43, 44, 48, 57]). The figures 1 and 2 give two examples of these representational schemes. A major characteristic of these representational schemes is that the low–level one is well suited for the precise and deterministic handling of the connectivity patterns of small networks, whereas the high–level one is well suited for the handling of the structural regularities of large networks. The low– and high–level schemes establish repesentational extremes between which many "mixed" genotypic representations are possible.

In view of genotypic representation of ANNs it is interesting to ask how and to what extend real brains are genetically encoded. Unfortunately, only little is known about this encoding. However, it is commonly agreed that the degree of brain determinism decreases from evolutionary lower to higher animals. Whereas this determinism is almost absolute in invertebrates, it allows great variability in vertebrates. Especially in mammals the brain development depends on both genetic and epigenetic factors, and requires the individual's interaction with its environment. See e.g. [18] for further details.

C. Hybrid Learning

All the approaches to a structural evolution of ANNs mentioned above implement the following **hybrid learning cycle**:

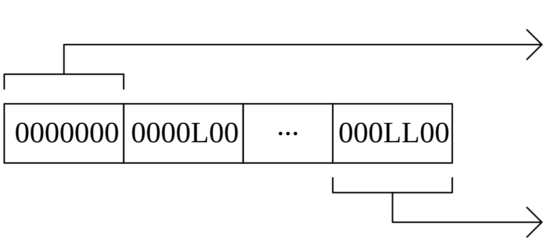
1. Creation of the next population of ANNs by means of mutation, recombination. and fitness–oriented selection. (The initial population is created at random.)
2. Training of the ANNs by conventional neural learning algorithms.
3. Evaluation of the fitness values of the ANNs with respect to some given design criteria.
4. If the desired result is obtained, then stop; otherwise goto step 1.

(In this cycle the genotypes and the phenotypes are not explicitly distinguished.) Of course, despite this uniform learning cycle, the approaches show great differences in detail. This concerns, in particular, the parent–offspring replacement strategy, the evolutionary operators and the neural learning procedures.

The price that has to be paid for using EAs for the structural network design is that of high computational costs. In order to cope with these costs (at least partially), one can employ the "natural parallelism" being inherent in these algorithms.

III. Evolutionary Network Training

Another way of synthesizing the fields of ANNs and EAs is to use EAs instead of standard neural learning algorithms for training ANNs; see e.g. [8, 12, 14, 25, 28, 37, 45, 47, 53, 62, 65, 66, 67]. The major idea underlying this synthesis is to interpret the weight matrices of the ANNs as genotypes, to change the weights by means of specific mutation and



	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	0	L	0	0
3	L	0	0	0	0	L	0
4	L	L	0	0	0	0	0
5	0	L	0	0	0	0	0
6	0	0	L	L	0	0	0
7	0	0	0	L	L	0	0

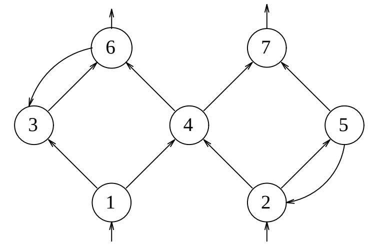


FIGURE 1: *Example of a low–level network representation according to [40]. The figure shows the genotypic representation (left) of the connectivity matrix (middle) of a simple neural net (right). The matrix entry (i,j) specifies the type of constraint on the connection from unit j to unit i; thus, row i of the matrix represents the constraints on the connections to unit i, and column j represents the constraints on the connections from unit j. Entry "0" means "weight fixed at zero", and entry "L" means "learnable".*

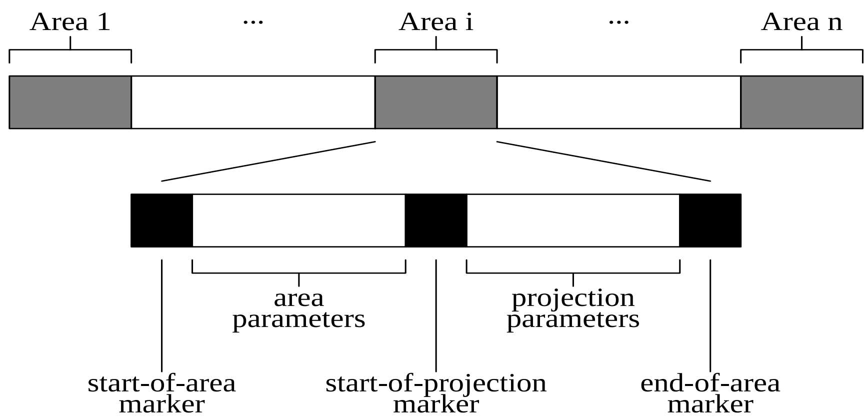


FIGURE 2: *Example of a high–level network representation according to [24]. A network "blueprint" is genotypically represented by a bit string that consists of several segments. Each segment specifies* (i) *the structure (e.g., the number of units and their spatial arrangement) of some area of the network by means of area parameters and* (ii) *the connections (e.g., their density and projection field) from this area to other areas by means of projection parameters.*

recombination operators, and to use the error produced by the ANNs as the fitness measure which guides selection. This leads to the following **evolutionary training cycle**:

1. Creation of the next population of ANNs by means of mutation, recombination and fitness–oriented selection of the the weight matrices. (The initial population is randomly created.)
2. Evaluation of the fitness values of the ANNs.
3. If the desired result is obtained, then stop; otherwise goto step 1.

Similar to the evolutionary–design approaches, the evolutionary–training approaches show great differences in detail; this concerns, in particular, the genotypic representation of the weight matrices (e.g., binary–string [66] versus real–number encoding [45]).

Perhaps the most striking argument for evolutionary–training approaches is that they, in contrast to the standard gradient–descent neural learning algorithms, inherently tend to avoid to get stuck in local minima of the error surface over the weight space. Evolutionary training was successfully applied to tasks like the XOR/424-encoder/adder problems, the construction of networks that approximate functions, categorization, robot–arm positioning, and pole balancing. However, despite these successes it is still unclear how evolutionary and standard neural training compare with each other; see e.g. [45, 3, 34] for some controversy statements and results. Although these results indicate a superiority of neural learning algorithms in general, in particular cases the outcome of this comparison seems to depend on the error surface and the task to be learnt by the ANN. More investigations are necessary to clarify this point.

IV. Other Hybrid Approaches

This section briefly overviews other hybrid approaches which have been done at the intersection of ANNs and EAs, but do not fall into the categories "evolutionary design" and "evolutionary training".

In [64] an EA was used for pruning unnecessary connections of ANNs after they have been trained by a standard neural learning algorithm. The idea was to evolve networks that are smaller than the initial ones, but still realize the desired tasks. In [5, 31] it is described how critical parameters of standard neural learning algorithms (e.g., learning rate and momentum in the case of backpropagation) can be optimized by means of EAs. In [5, 33] it is shown how the global sampling property of EAs can be combined with the local search performed by conventional neural learning algorithms. The idea underlying this interesting synthesis is to use EAs for searching the initial weight space of ANNs, and the to proceed with conventional neural gradient–descent learning methods. In [21] the evolutionary concepts of mutation and selection have been employed as tools for escaping from local error minima of backpropagation–trained networks, and in [35] these concepts have been used to enable unsupervised learning networks to change their structure adaptively. A survey of formal models for describing the dynamics of genotype–phenotype evolution is provided in [46]. Other interesting works are described in [6, 7, 10, 17, 23]; here it is investigated how simple neural learning rules on their own, instead of weight matrices or network structures, can be evolved with the help of EAs.

Finally, there are two other important questions concerning the relation between learning and evolution that have been addressed by several researchers in the context of ANNs. First, the question under which conditions and how quickly learning can evolve; see [39, 55, 56]. Here a computational model was described which puts light on the trade–off between the necessity to evolve learning and the ability to learn during lifetime. And second, the question how learning can guide evolution; see e.g. [4, 27, 47]. Here the underlying argumentation (known as the Weismann Doctrine in evolutionary theory and biology) is that the ability to learn influences the adaptability and, with that, the number of descendants of an individual; and this, in turn, leads to a modified evolutionary search space.

V. Concluding Remarks

The intersection of ANNs and EAs establishes a very challenging research field for two opposite reasons. On the one side, this research field offers the opportunity to obtain results and insights which are useful and profitable for both the area of ANNs and the area of EAs. For instance, from the point of view of the ANNs, a better understanding can be gained with respect to the structure–function connection of networks and the optimal parameter tuning of standard neural learning procedures; and from the point of view of the EAs, a better understanding can be gained with respect to the representation of genotypes and the choice of evolutionary operators. The hybrid approaches developed so far and overviewed in this paper do the first steps in these directions.

On the other side, this research field is a very difficult one, simply because it brings together unsolved problems from two complex areas. For that reason, and in order to obtain results which are really expressive and useful, "hybrid work" has to be done extremely carefully. This was not always the case in the past; for instance, some experimental results obtained so far and reported in the literature are too specific or simply too vague to lead to new insights. There are many open questions existing and arising in this field, and a lot of experimental and theoretical efforts are required in order to answer them. These efforts might be greatly supported and inspired by taking related work from other disciplines like genetics or the neurosciences into consideration (see e.g. [59, 61]).

To summarize this conclusion: work at the intersection of ANNs and EAs is really worth to be done, and is worth to be done carefully.

References

[1] Albrecht, R. F., Reeves, C. R., & Steele, N. C. (Eds.) (1993). *Artificial neural nets and genetic algorithms. (Proceedings of the International Conference in Innsbruck, Austria)*. Springer.

[2] Bäck, T., Hoffmeister, F., & Schwefel, H.-P. (1992). *Applications of evolutionary algorithms.* Techn. Rep. No. SYS–2/92. Department of Computer Science, University of Dortmund, Germany.

[3] Bartlett, P., & Downs, T. (1990). *Training a neural network with a genetic algorithm.* Techn. Report. Department of Electrical Engineering, University of Queensland.

[4] Belew, R. K. (1989). *Evolution, learning and culture: Computational metaphors for adaptive algorithms.* CSE Techn. Rep. CS89–156. Computer Science and Engineering Department, University of California, San Diego.

[5] Belew, R. K., McInerney, J., & Schraudolph, N. N. (1990). *Evolving networks: Using the genetic algorithm with connectionist learning.* CSE Techn. Rep. CS90–174. Computer Science and Engineering Department (C–014), University of California, San Diego (La Jolla).

[6] Bengio, S., & Bengio, Y. (1991). *Learning a synaptic learning rule.* Techn. Rep. No. 751. Départment d'Informatique et de Recherche Operationelle, Université de Montréal.

[7] Bengio, S., Bengio, Y., Cloutier, J., & Gecsei, J. (1993). *Generalization of a parametric learning rule.* Départment d'Informatique et de Recherche Operationelle, Université de Montréal.

[8] Caudell, T. P., & Dolan, C. P. (1989). Parametric connectivity: Training of constrained networks using genetic algorithms. In *Proc. of the Third Intl. Conf. on Genetic Algorithms* (pp. 370–374).

[9] Camargo, F.A. (1991). *Learning algorithms in neural networks.* Technical Report. Computer Science Department, Columbia University.

[10] Dasdan, A., & Oflazer, K. (1993). *Genetic synthesis of unsupervised learning algorithms.* Department of Computer Engineering and Information Science, Bilkent University, Ankara.

[11] Davis L. (Ed.) (1991). *Handbook of Genetic Algorithms.* Van Nostrand Reinhold.

[12] de Garis, H. (1990). Genetic programming. Modular neural evolution for Darwin machines. In *Proc. of the 1990 Intl. Joint Conf. on Neural Networks* (Vol. I, pp. 194–197).

[13] de Garis, H. (1990). Brain building. Building artificial nervous systems with GenNet modules. In *Proc. of the First Intl. Workshop on Problem Solving from Nature.*

[14] Dodd, N. (1990). Optimisation of network structure using genetic techniques. In *Proc. of the 1990 Intl. Joint Conf. on Neural Networks.*

[15] Dolan, C. & Dyer, M. G. (1987). *Symbolic schemata in connectionist memories: role binding and the evolution of structure.* Techn. Rep. UCLA–AI–87–11. Computer Science Department, University of California, Los Angeles.

[16] Dolan, C., & Dyer, M.G. (1987). Towards the evolution of symbols. In *Proc. of the Second Intl. Conf. on Genetic Algorithms* (pp. 769–775).

[17] Fontanari, J.F., & Meir, R. (1991). Evolving a learning algorithm for the binary perceptron. *Network*, 2, 353–359.

[18] Gierer, A. (1988). Spatial organization and genetic information in brain development. In *Biological Cybernetics*, 59, 13–21.

[19] Goldberg, D.E. (1989). *Genetic algorithms in search, optimization and machine learning.* Addison–Wesley.

[20] Gruau, F. (1992). *Encoding of genetic neural networks.* Rep. No. 92-1. Ecole Normale Superieure de Lyon, France.

[21] Hagiwara, M., & Nakagawa, M. (1989). Supervised learning with artificial selection. In *Proc. of the 1989 Intl. Joint Conf. on Neural Networks* (Vol. II, p. 611).

[22] Hancock, P. J. B. (1990). *GANNET: Design of a neural net for face recognition by genetic algorithm.* Techn. Report. Departments of Computing Science and Psychology, University of Stirling.

[23] Hancock, P.J.B., Smith, L.S., & Phillips, W.A. (1991). A biologically supported error–correcting learning rule. In T. Kohonen et.al. (Eds.), *Proc. of the Intl. Conf. on Artificial Neural Networks – ICANN91* (Vol. 1, pp. 531–536). Elsevier.

[24] Harp, S.A., Samad, T., & Guha, A. (1989). Towards the genetic synthesis of neural networks. In *Proc. of the Third Intl. Conf. on on Genetic Algorithms* (pp. 360–369).

[25] Heistermann, J. (1990). Learning in neural nets by genetic algorithms. In R. Eckmiller, G. Hartmann & G. Hauske (Eds.), *Parallel processing in neural systems and computers* (pp. 165–168). Elsevier.

[26] Hinton, G.E. (1989). Connectionist learning procedures. *Artificial Intelligence*, 40, 185–234.

[27] Hinton, G.E. & Nowlan, S.J. (1987). How learning can guide evolution. In *Complex Systems*, 1, 495–502.

[28] Höffgen, K.–U., & Siemon, H. P. (1990). *Approximation of functions with feedforward networks.* Techn. Rep. no. 346. Department of Computer Science, University of Dortmund, Germany.

[29] Hoffmeister, F., & Bäck, T. (1990). Genetic algorithms and evolution strategies: Similarities and differences. In *Proc. of the First Intl. Workshop on Problem Solving from Nature.*

[30] Holland, J.H. (1975). *Adaptation in natural and artificial systems.* The University of Michigan Press.

[31] Jacobs, R.A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1, 295–307.

[32] Jones, A. J. (1993). Genetic algorithms and their applications to the design of neural networks. *Neural Computing & Applications*, 1, 32–45.

[33] Kitano, H. (1990). Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4, 461–476.

[34] Kitano, H. (1990). Empirical studies on the speed of convergence of neural network training using genetic algorithms. In *Proc. of the Eighth National Conf. on Artificial Intelligence.*

[35] Lee, T.–C., & Peterson, A. M. (1989). *SPAN: A neural network that grows.* Techn. Report. Department of Electrical Engineering, Stanford University.

[36] Lehar, S., & Weaver, J. (1987). A developmental approach to neural network design. In *Proc. of the IEEE Intl. Conf. on Neural Networks* (Vol.I, pp.97–104).

[37] Merrill, J. W. L.. & Port, R. F. (1988). *A stochastic learning algorithm for neural networks.* Techn. Rep. no. 236. Departments of Linguistics and Computer Science, Indiana University.

[38] Merrill, J. W. L., & Port, R. F. (1988). *Fractally configured neural networks.* Techn. Rep. no. 249. Computer Science Department, Indiana University.

[39] Miller, G. F., & Todd, P. M. (1990). Exploring adaptive agency I: Theory and methods for simulating the evolution of learning. In D. S. Touretzky, J. L. Elman, T. J. Sejnowksi & G. E. Hinton (Eds.), *Proc. of the 1990 Connectionist Models Summer School.*

[40] Miller, G. F., Todd, P. M., & Hedge, S. U. (1989). Designing neural networks using genetic algorithms. In *Proc. of the Third Intl. Conf. on Genetic Algorithms* (pp. 379–384).

[41] Mjolsness, E., & Sharp, D. H. (1986). A preliminary analysis of recursively generated networks. In J. Denker (Ed.), *Neural networks for computing* (pp. 309–314). Snowbird.

[42] Mjolsness, E., Sharp, D. H., & Alpert, B. K. (1987). Recursively generated neural networks. In *Proc. of the IEEE Intl. Conf. on Neural Networks* (Vol. III, pp. 165–171).

[43] Mjolsness, E., Sharp, D. H., & Alpert, B. K. (1988). *Scaling, machine learning and genetic neural nets.* Techn. Rep. LA–UR–88–142. Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM.

[44] Mjolsness, E., Sharp, D. H., & Reinitz, J. (1990). *A connectionist model of development.* Research Rep. YALEU/DCS/RR–796. Department of Computer Science, Yale University.

[45] Montana, D. J., & Davis, L. (1989). *Training feedforward neural networks using genetic algorithms.* Techn. Report. BBN Systems and Technologies, Inc., Cambridge, MA.

[46] Mühlenbein, H., & Kindermann, J. (1989). The dynamics of evolution and learning – Towards genetic neural networks. In R. Pfeifer, Z. Schreter, F. Fogelman-Soulie & L. Steels (Eds.), *Connectionism in perspective.* Elsevier.

[47] Nolfi, S., Elman, J. L., & Parisi, D. (1990). *Learning and evolution in neural networks.* CRL Techn. Rep. 9019. Center for Research in Language, University of California, San Diego.

[48] Nolfi, S., & Parisi, D. (1991). *Growing neural networks.* Techn. Rep. PCIA–91–15. Department of Cognitive Processes and Artificial Intelligence, C.N.R. Rome, Italy.

[49] Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Frommberg–Holzboog.

[50] Rudnick, M. (1990). *A bibliography of the intersection of genetic search and artificial neural networks.* Techn. Rep. No. CS/E 90–001. Oregon Graduate Center, Beaverton, OR.

[51] Rumelhart, D.E., McClelland, J.L., and the PDP Research Group (1986). *Parallel distributed processing. Explorations in the microstructure of cognition.* Vol.s 1 & 2. The MIT Press.

[52] Schiffmann, W., & Mecklenburg, K. (1990). Genetic generation of backpropagation trained neural networks. In R. Eckmiller, G. Hartmann & G. Hauske (Eds.), *Parallel processing in neural systems and computers* (pp. 205–208). Elsevier.

[53] Scholz, M. (1990). A learning strategy for neural networks based on a modified evolutionary strategy. In *Proc. of the First Intl. Workshop on Problem Solving from Nature.*

[54] Schwefel, H.–P. (1977). *Numerische Optimierung von Computermodellen mittels Evolutionsstrategie.* Birkhäuser. (English Version: 1981, *Numerical optimization of computer models*, Wiley.)

[55] Todd, P. M., & Miller, G. F. (1990). Exploring adaptive agency II: Simulating the evolution of associative learning. In *Proc. of the First Intl. Workshop on Problem Solving from Nature.*

[56] Todd, P. M., & Miller, G. F. (1991). Exploring adaptive agency III: Simulating the evolution of habituation and sensitization. In *Proc. of the First Intl. Workshop on Problem Solving from Nature.*

[57] Voigt, H.-M., Born, J., & Santibáñez-Koref, I. (1993). *Evolutionary structuring of artificial neural networks.* Techn. Report. Bio– and Neuroinformatics Research Group, Technical University Berlin, Germany.

[58] Weiß, G. (1990). *Combining neural and evolutionary learning: Aspects and approaches.* Techn. Rep. FKI–132–90. Institut für Informatik, Technische Universität München.

[59] Weiß, G. (1993, to appear). Towards the synthesis of neural and evolutionary learning. In O. Omidvar (Ed.), *Progress in neural networks* (Vol. 5, Chapter 5). Ablex Publ.

[60] Weiß, G. (1993, in preparation). *Applying evolutionary algorithms to the structural design of artificial neural networks.* Techn. Report. Institut für Informatik, Technische Universität München.

[61] Weiß, G. (1993, submitted to IEEE World Congress on Computational Intelligence). Neural networks and evolutionary computation. Part II: Hybrid approaches in the neurosciences.

[62] Whitley, D. (1988). *Applying genetic algorithms to neural net learning.* Techn. Rep. CS–88–128. Department of Computer Science, Colorado State University.

[63] Whitley, D. (1989). The GENITOR algorithm and selective pressure: Why rank–based allocation of reproductive trials is best. In *Proc. of the Third Intl. Conf. on Genetic Algorithms* (pp. 116–121).

[64] Whitley, D., & Bogart, C. (1990). The evolution of connectivity: Pruning neural networks using genetic algorithms. In *Proc. of the Intl. Joint Conf. on Neural Networks* (Vol. I, pp. 134–137).

[65] Whitley, D., & Hanson, T. (1989). Optimizing neural networks using faster, more accurate genetic search. In *Proc. of the Third Intl. Conf. on Genetic Algorithms* (pp. 391–397).

[66] Whitley, D., Starkweather, T., & Bogart, C. (1989). *Genetic algorithms and neural networks: Optimizing connections and connectivity.* Techn. Rep. CS–89–117. Department of Computer Science, Colorado State University.

[67] Wieland, A. P. (1990). Evolving controls for unstable systems. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski & G. E. Hinton (Eds.), *Proc. of the 1990 Connectionist Models Summer School* (pp. 91–102). Morgan Kaufmann.

[68] Wilson, S. W. (1989). Perceptron Redux: Emergence of structure. In S. Forrest (Ed.), *Emergent computation* (pp. 249–256). MIT/North-Holland.

[69] Yao, X. (1992). *A review of evolutionary artificial neural networks.* Techn. Report. Commonwealth Scientific and Industrial Research Organisation, Victoria, Australia.