Generating complex connectivity structures for large-scale neural models

Hülse, Martin

Publication date:
2008

Citation for published version (APA):
Hülse, M. (2008). *Generating complex connectivity structures for large-scale neural models*. 849.

Generating complex connectivity structures for large-scale neural models

Martin Hülse

Department of Computer Science, Aberystwyth University
Aberystwyth, Ceredigion, Wales, SY23 3DB, UK
msh@aber.ac.uk
http://users.aber.ac.uk/msh

Abstract. Biological neural systems and the majority of other real-world networks have topologies significant different from fully or randomly connected structures, which are frequently applied for the definition of artificial neural networks (ANN). In this work we introduce a deterministic process generating strongly connected directed graphs of fractal dimension having connectivity structures very distinct compared with random or fully connected graphs. A sufficient criterion for the generation of strongly connected directed graphs is given and we indicate how the degree-distribution is determined. This allows a targeted generation of strongly connected directed graphs. Two methods for transforming directed graphs into ANN are introduced. A discussion on the importance of strongly connected digraphs and their fractal dimension in the context of artificial adaptive neural systems concludes this work.

1 Introduction

Within the context of neural computation and cognitive science artificial neural networks (ANN) are frequently utilized as the basic building blocks for large-scale neural models in order to explore the nature of complex information processing exploited in animals and human beings. The majority of such neural models are based on connectivity structures which match with the classical types of ANN, such as, Multi-layered-perceptrons, Hopfield- or Elman-networks [5, 7]. All these network types establish a connectivity structure close to fully connected networks. The application of fully connected networks, however, might become crucial with respect to plausibility if they are intended to model biological systems. Fully connected ANN can hardly represent brain-like neural structures, if, as only one example, approximately 10^{11} neurons in the human brain are coordinated by "only" 10^{15} synapses [10].

An alternative, in particular for large-scale neural models, to overcome fully connected neural networks is the creation of random graph structures [1]. Nevertheless, random graph models do not well describe some essential properties of real-world or biological networks, such as degree-distribution [13]. Therefore, we argue, while modeling large-scale neural networks one must consider alternatives for the projections between neural assemblies; alternatives which go beyond random graphs and fully connected structures.

Furthermore, large-scale neural models are applied for robot control more and more [8, 16]. Such implementations on autonomous robots might be motivated as a proof of concept as well as for targeting specific issues of embodiment [14]. However, autonomous robots have usually very limited computational resources CPU and memory. Hence, for performance reasons it becomes important to utilize highly connected networks established by as less connections as possible.

The objective of this paper is to introduce a deterministic method which enables us to create *strongly connected directed graphs* [15] established by a number of edges magnitudes smaller than in fully connected graphs / networks. The generation process is inspired by fractal sets, namely Sierpiński carpets [12]. This makes the resulting structures very distinct compared with random and fully connected graphs. Due to the simplicity and deterministic character of the generation process, this method seems to be a promising alternative for the targeted generation of directed graphs in general as well as it opens a wide field for applications in many areas of neural modeling. We will introduce two strategies which allow alternative definitions of connectivity structures for feedforward and recurrent neural networks.

2 Directed graphs, Sierpiński carpets, and strongly connected digraphs of fractal dimension

In this section we demonstrate how Sierpiński carpets motivate a process for a targeted generation of strongly connected directed graphs (digraphs), which will lead us to the definition of *digraphs of fractal dimension.*

2.1 Directed graphs and Sierpiński carpets

A directed graph or a digraph is a pair $G(V, E)$, where V is a set of vertices (sometime also called nodes) and E is a subset of $V \times V$. An edge $e_{ij} \in E$ represents an edge from node v_i to v_j, where e_{ij} is the in-coming edge for v_j and the out-going for v_i. In the following we also allow edges e_{ii} and therefore, edges can be in-coming as well as out-going edge for one and the same node. We call the number of in-coming edges of node v_i in-degree $d_{in}(v_i)$ and the number of out-going edges the out-degree $d_{out}(v_i)$. In a digraph the number of in-coming edges is equal to the number of the out-going: $\sum_i^n d_{in}(v_i) = \sum_i^n d_{out}(v_i)$, where n is the number of nodes in G. A directed path between node v_s and v_t in G is a sequence of edges $e_1, e_2, \ldots e_k$ such that the end node of edge e_i is the start node of $e_{(i+1)}$, $i = 1, 2 \ldots k$. If there exists a directed path between each pair of nodes in a digraph, we call it *strongly connected.*

The structure of a digraph $G(V, E)$ can be represented by an adjacency matrix M. Each matrix element m_{ij} of an adjacency matrix can either be zero or one. The element m_{ij} is one, if and only if $e_{ij} \in E$.

In the following section we describe how adjacency matrixes can build a bridge between Sierpiński carpets and digraphs as well as they give us a process for the deterministic development of strongly connected digraphs.

Sierpiński carpets result from an iterative process where a pattern is successively used to replace specific regions in an evolving pattern. In Fig. 1 an example is shown which illustrates this process. One starts with a given pattern, here a square divided into 3×3 equal sub-squares. The sub-squares are labeled either black or white, which creates a specific pattern. In each iteration the original pattern is used to replace all the black labeled regions, by this pattern again. This leads to a finer partition of the originally given pattern. Thus, after the first iteration, when each black labeled square is replaced by the pattern, we have a square subdivided into $9 \times 9 = 81$ regions, instead of the 9 regions given in the original pattern. As we see, after 5 iterations we have a "fractal set" represented on a square, regularly subdivided into 729×729 equal sub-squares. For infinity iterations we get the Sierpiński carpet, that is a set of fractal dimension.

In this process towards a fractal set we have to distinguish between the pattern $\mathcal{P}$ which is transformed into a new pattern $\mathcal{P}'$ and the pattern which determines this transformation. The latter we call *mask* $\mathcal{M}$. In the following we only allow masks and patterns with dimension $n \times n$ $(n > 1)$, where n indicates the segmentation. Obviously, a mask of segmentation S, written as $\mathcal{M}_S$, applied to a pattern $\mathcal{P}$ of segmentation m results in a new pattern $\mathcal{P}'$ of segmentation $S \cdot m$.

We utilize this transformation process to generate digraphs simply by interpreting the resulting patterns as an adjacency matrix. Namely, the black labeled sub-squares are interpreted as edges, i.e. black color represents value 1 in the corresponding adjacency matrix, while white squares indicate the zero entries, i.e. no edge. In this way a pattern or a mask of segmentation S is transformed into an adjacency matrix representing a digraph of S nodes and k edges, where k is the number of black labeled sub-squares in the pattern. Examples of 3×3 patterns transformed into digraphs are given in Figure 1.

With respect to adjacency matrixes we see that the Kronecker product [6] can be applied in order the define an algorithm which is isomorph to the process generating Sierpiński carpets:

$$\begin{aligned} D_0 &:= M_S \\ D_{n+1} &:= D_n \otimes M_S, \end{aligned}$$

where M_S is an adjacency matrix $(M(i,j) \in \{0;1\})$ of dimension $S \times S$. In the following we refer to this algorithm as the *digraph generating process* DGP.

Due to the direct interpretation of patterns and masks as adjacency matrixes and digraphs we make use of these terms synonymously. The only important thing here is that a mask, either written as pattern $\mathcal{M}_S$ or adjacency matrix M_S, is the only seed which initializes the DGP and therefore completely determines the resulting connectivity structure.

2.2 Masks creating strongly connected digraphs

We now ask which masks of a given segmentation S create strongly connected digraphs. For the investigation of this question we start with the simplest form

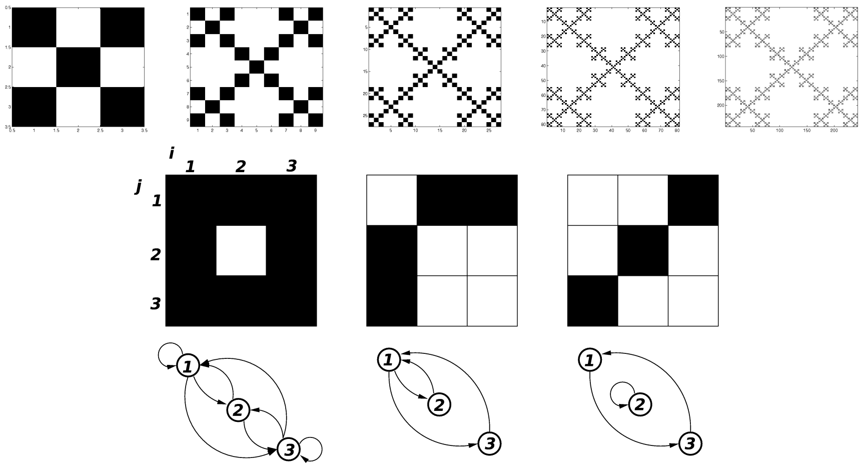

Fig. 1. Top: An example of the first five iterations towards Sierpiński carpets. After 5 iterations the original given 3 × 3 partition is transformed into a 729 × 729 partition, which can be interpreted as adjacency matrix for a digraph containing 729 nodes. See text for explanation. **Bottom:** Examples of digraphs derived from 3 × 3 patterns.

of strongly connected digraphs: cycles, also called rings. Each strongly connected digraph $G(V, E)$ containing as many edges as nodes forms a cycle. Consequently such a digraphs has no loops (also called self-connections), i.e. $e_{ii} \notin E$. Further more, each node has only one in-coming and one out-going edge.

Assume the general case where we have a mask M_S and a digraph of segmentation T represented by the adjacency matrix D_0. Both, M_S and D_0, are digraphs forming cycles. Due to the digraph generation process we can deduce, that $D_1 := D_0 \otimes M_S$:

1. has as many nodes as edges $(T \cdot S)$,
2. only contains nodes with one out-going and one-incoming edge, and
3. has no node with a self-connection (loop).

Therefore, D_1 forms either a cycle or is fallen apart into "sub-cycles". In the latter case D_1 wouldn't be strongly connected anymore.

Whether D_1 is a cycle or not is actually determined by the number of nodes in M_S *and* D_0, in other words it depends on their segmentation. The cycles which M_S and D_0 are forming can be represented by the corresponding sequence of nodes. Taking the example shown in Fig. 2 we get for the following sequence for D_0:

$$(1^*) \rightarrow (3^*) \rightarrow (2^*) \rightarrow (4^*) \rightarrow (1^*),$$

for M_S we have:

$$(1^+) \rightarrow (3^+) \rightarrow (2^+) \rightarrow (1^+),$$

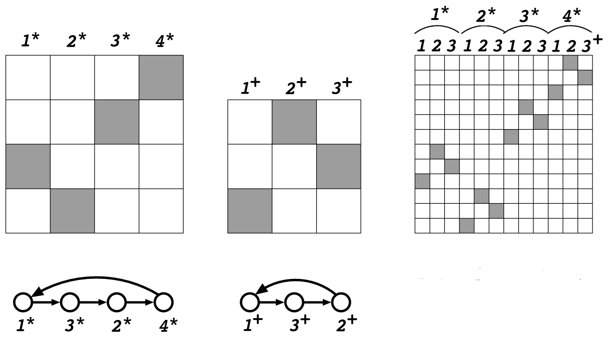

Fig. 2. Example of a process where a cycle with 4 nodes (D_0) is transformed by a mask (M_S, cycle with 3 nodes) into a new cycle of 12 nodes, D_1. See text for details.

and D_1 is represented by:

$$(1^*, 2^+) \rightarrow (3^*, 1^+) \rightarrow (2^*, 3^+) \rightarrow (4^*, 2^+) \rightarrow (1^*, 1^+) \rightarrow (3^*, 3^+) \rightarrow \ldots$$

In order to relate these two sequences to the sequence of the resulting graph D_1 we apply a numbering for D_1 which somehow preserves the numbering of D_0 and M_S. Fig. 2 shows how this can be achieved. Each node in D_1 is now represented by a number of two components (a^*, b^+). Starting with an arbitrary node (a^*_1, b^+_1) we can now write down the sequence of nodes forming a directed path in the following general form:

$$(a^*_1, b^+_1) \rightarrow (a^*_2, b^+_2) \rightarrow \ldots \rightarrow (a^*_k, b^+_k) \rightarrow (a^*_1, b^+_1).$$

Note, due to the definition of DGP each node in D_1 can have only one successor, which is not the node itself. Therefore, we can conclude: if $k = S \cdot T$ then we have a cycle formed by $S \cdot T$ nodes, which is the number of nodes in D_1. Thus, it would follow: D_1 is still strongly connected. On the other hand, if $k < S \cdot T$ then D_1 consist of sub-graphs forming cycles and therefore D_1 isn't neither strongly connected nor connected at all.

We see, it is the value of k which indicates whether or not the result is strongly connected. However, the crucial point for calculating k is to understand how the two sequences a^*_i and b^+_i are determined by D_0 and M_S. In fact, the sequence for a^*_i (first component of the D_1 numbering) is exactly the same of D_0. No matter how the sequence of D_1 actually looks like, considering only the a^*-sequence, we see the same sequence as for D_0:

$$(1^*, b^+_t) \rightarrow (3^*, b^+_{t+1}) \rightarrow (2^*, b^+_{t+2}) \rightarrow (4^*, b^+_{t+3}) \rightarrow (1^*, b^+_{t+4}) \rightarrow \ldots$$

The same holds for the b^+_i-sequence (second component of the D_1 numbering). It is determined by the "cycle sequence" of M_S:

$$(a^*_t, 1^+) \rightarrow (a^*_{t+1}, 3^+) \rightarrow (a^*_{t+2}, 2^+) \rightarrow (a^*_{t+3}, 1^+) \rightarrow \ldots$$

Combining these two observation we get: $k = S \cdot T$ if, and only if S is not divisor of T or vice versa. In other words D_1 remains strongly connected. While for the other case, i.e. S is divisor of T or vice versa, then we get $k = max(S, T)$, meaning D_1 consist of $min(S, T)$ sub-graphs with k nodes forming separated cycles and therefore D_1 isn't connected at all.

This result tells us that DGP doesn't create strongly connected digraphs if it is initialized with a cycle, because D_0 and M_S have the same segmentation per definition.

However, examining the same argumentation we see that by extending M_S with a single loop (i.e. one self-connection) all separated sub-cycles will be connected. An additional self-connection operates like a junction connecting all separated sub-cycles. In this case the resulting digraph D_1 remains strongly connected. Further more, all resulting digraphs $D_n \ \ (n > 1)$ would also be strongly connected, because, due to the definition of DGP, the given mask (with its self-connection) is always applied to a strongly connected digraph. Speaking precisely, the mask is always applied to a digraph containing a sub-graph forming a cycle which involves all nodes.

At this point we are able to formulate a sufficient criterion which guarantees strongly connected graphs for the DGP: *If M_S represents a cycle with at least one self-connection then all the resulting digraphs D_n are strongly connected.* In the following we call such masks *complete*.

We also see that each mask containing a complete mask as sub-structure will generate strongly connected digraphs as well.

2.3 Digraphs of fractal dimension and their degree distribution

The generation of strongly connected digraphs therefore has always to start with a mask $\mathcal{M}_S$ containing at least $S + 1$ edges. The maximal number of edges in M_S is S^2 and it is easy to see that such a mask generates only fully connected digraphs. Hence, the non-trivial cases of connected graphs are generated by masks with n edges, where $S < n < S^2$. Interestingly enough, masks with this number of labeled segments generate Sierpiński carpets of fractal dimensions d_f between 1 and 2 [12], since: $d_f = \frac{log(n)}{log(S)}$, from which follows: $1 < d_f < 2$. Therefore, we say a *fractal digraph* or a *digraph of fractal dimension d_f* is defined as a strongly connected digraph resulting from a mask of fractal dimension d_f.

The degree distribution is an important property in order to classify networks. Due to the deterministic nature of the DGP the mask determines this distribution in the following way. Be M_S the adjacency matrix. Out- and in-degree for each node in the digraph are directly given by the sums over the entries in the column or row of M_S:

$$d_{in}(v_n) = \sum_{j=1}^{S} M_S(n, j), \quad d_{out}(v_n) = \sum_{j=1}^{S} M_S(j, n), \qquad 1 \le n \le S.$$

Considering only the in-degree we can calculate the degree for each node in digraph D_i (i.e. resulting digraph after i iterations of DGP initialized with M_S

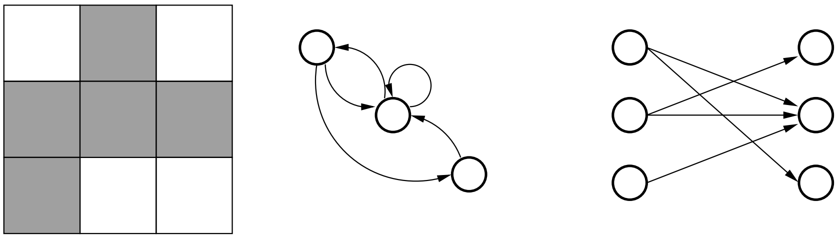

Fig. 3. Two ways of transforming a given digraph of fractal dimension (left) into a artificial neural networks. First, the digraph is directly interpreted as ANN with recurrent neural connections (middle). Second, the adjacency matrix as a description of a feed-forward network (right).

and $i \geq 0$) as follows:

$$(d_{in}(v_1) + d_{in}(v_2) + \ldots + d_{in}(v_S))^{i+1}$$

Solving this equation in a symbolic manner we get S^{i+1} products each representing the in-degree of one node in D_i. The distribution of these product values is the actual distribution of the in-degrees in D_i.

As an example let us assume we have a complete mask where only one node has an in- and out-degree greater 1, for all the other nodes in- and out-degree is one. The resulting digraphs will have a degree distribution characteristic of scale-free networks. Thus, the majority of nodes have very less in-coming and out-going edges but a few nodes (usually called hubs) have degrees magnitudes larger then the average [2]. On the other hand we can create digraphs out of masks, where each node has the same number of in- and out-degree. The resulting digraphs are going to have equal in- and out-degree as well. Due to this relation of the degree distribution between mask and resulting digraphs one is able to generate networks of specific degree-distribution.

3 From fractal digraphs to artificial neural networks

There are in principle two strategies to turn a digraph of fractal dimension into an artificial neural network (see Figure 3). First, the adjacency matrix / the digraph can directly be interpreted as a neural network containing recurrences of any kind. Second, the adjacency matrix can purely be seen as the connections between two separated layers of neurons: input and output layer. Both layers contain the same number of neurons. In this way a feed-forward structure between two neuron layers is created. One can also think about a chain of feedforward connections where each projection layer might be based on different digraphs of fractal dimension.

The recurrent case might be interesting as method for the generation of reservoirs of non-linear dynamics. Based on random graphs, this has been done

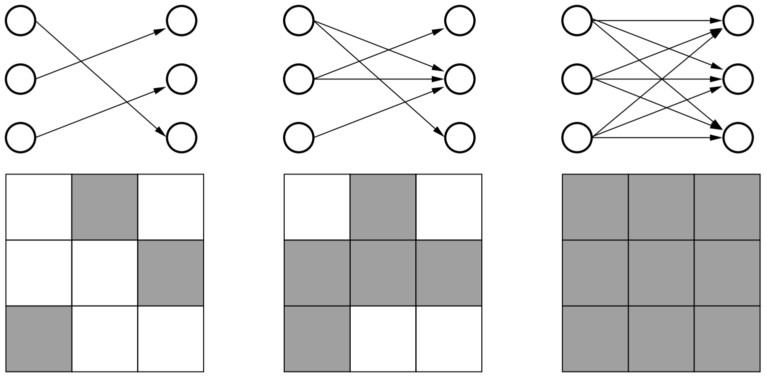

Fig. 4. Three examples of feedforward connections between two neural layers of an ANN. The left shows a non-degenerated matching between input and output signal. Each neuron in the left layer does only activate one neuron in the right layer. Due to the introduced representation this can be described as a cycle. On the right a completely degenerated matching. Each neuron on the left activates each neuron on the right. This is represented by a fully connected digraph. In between a degenerated matching formed by a digraph of fractal dimension.

in the echo-state [9] and liquid-state-machine [11] approach. The intention of using digraphs of fractal dimension as dynamical reservoir is one reason for us to aim for strongly connected digraphs only. The dynamics of echo-state and liquid-state-machines rely on the recurrences. If an underlying digraph would be only connected (i.e. not strongly connected) neurons can emerge which only project signals out of the reservoir or which would feed constant signals into it. This is obviously not the intention of a dynamical reservoir for both approaches. In the worst case a connected digraph could have no recurrences at all and therefore no complex dynamics would emerge. Only strongly connected digraphs guarantee recurrent neural structures involving all components of the network.

For feedforward structures the use of strongly connected digraphs is also essential. Strongly connected digraphs guarantee that a signal feed into an arbitrary node can be propagated through several layers to any other node. Assume a multi-layered network structure and each projection between the layers is based on the same digraph of fractal dimension. In this case we know there must exist a finite number of layers between the input and output layer which guarantees that each neuron of the input layer has at least one path to each neuron in the output layer. In theory the number of layers cannot be larger than the number of nodes in the underlying strongly connected digraph. It is not shown here, but simulations indicate that the mean value of the shortest path scales with $log(n)$ (n number of nodes). Hence, the number for layers supporting a signal flow through all network components scales with log either.

ANN with feedforward structures based on fractal digraphs might become an object of investigation within the Neural Darwinism approach to the function of the brain introduced, developed and promoted by Edelman [3]. According to this approach, an essential element for the brain-function is the matching between specific signal configurations and neural groups, which respond in a specific manner. Obviously, this matching must be sufficient specific in order to allow distinction among different signals, called recognition. However, more important within the Neural Darwinism approach is the argumentation, that such a matching must be degenerated. The assumption is, that there is more than one way to recognize a signal, that is, one signal configuration activates different neural groups as well as one neural group can be activated by different signal configurations. Two extremes of degeneration can be distinguished: a non-degenerated (unique) matching on one side and the completely degenerated matching on the other side. The Neural Darwinism approach claims that the variability of brain functions occurs within a neural organization is, somehow, located between these two extremes of non- and complete degeneration.

It is interesting to see that the introduced digraphs of fractal dimension create feedforward networks between these two extremes. The examples shown in Figure 4 are only simple schemas. However, it is not hard to imagine that the fractal dimension and degree distribution of a digraph determine the grad of degeneration. Therefore, we argue, that within the Neural Darwinism approach the introduced digraphs of fractal dimension might be a promising substrate for future research in order to model brain-like mechanisms of adaptation which take into account not only weight dynamics but also specific neural connectivity structures.

4 Conclusion

In this work we have introduced a process, called DGP, which allows the deterministic generation of strongly connected digraphs. This process is inspired by Sierpiński carpets. In order to apply this process for the development of connectivity structures for recurrent and feedforward structures of ANN we have formulated a sufficient criterion which guarantees the generation of strongly connected digraphs. The resulting digraphs are called digraphs of fractal dimension.

Furthermore, we have shown how the degree-distribution of the resulting digraphs is determined by the initial structure. This allows us a targeted generation of strongly connected digraphs with respect to the size (number of nodes) and the degree distribution of the network. We have indicated that strongly connected digraphs with scale-free network topologies can be expected to emerge for specific initializations. Thus, the introduced process is an efficient tool for the generation of a wide variety of connectivity structures for ANN establishing interactions beyond those provided by fully connected or random graphs.

Within the context of Neural Darwinism, we have highlighted the importance of ANN based on digraphs of fractal dimensions for the implementation of adaptive processes. Due to the fractal dimension, the resulting strongly connected networks are established by a number of edges magnitudes smaller then given in fully connected topologies. Hence, graphs of fractal dimension support the instantiation of highly connected ANN with much less computational costs. Therefore, the DGP might be an efficient tool for the instantiation of ANN on autonomous robot systems having limited computational resources. In addition to this, fractal digraphs within the context of cognitive robotics and "brain-based devices" might be become a promising method for a systematic investigation and modelling of biological neural systems and "their combinations of interactions that we don't fully understand yet" [4]. Our future work will be focused on this issue.

5 Acknowledgement

The author would like to thank Prof. Dr. Vassilis C. Mavron and Prof. Dr. Frank Pasemann for fruitful comments on the topic of this paper and gratefully acknowledges the support of EPSRC through grant EP/C516303/1.

References

1. Bollobás, B.: Random Graphs. Academic Press Inc., London, UK (1985)
2. Dorogovtsev, S. N., Mendes, J. F. F: Evolution of Networks: From Biological Nets to the Internet and WWW. Oxford University Press, Oxford, UK (2003)
3. Edelman, G. M.: Neural Darwinism. Oxford University Press, Oxford, UK (1989)
4. Edelman, G. M.: Learning in and from brain-based devices. Science 318(5853), 1103–1105 (2007)
5. Elman, J. L.: Finding structure in time. Cognitive Science, 14, 179–211 (1990)
6. Eves, H.: Elementary Matrix Theory. Dover publications (1980)
7. Hertz, J., Krogh, A., Palmer, R.G.: Introduction to the theory of neural computation. Perseus Books, USA (1991)
8. Ito, M., Noda, K., Hoshino, Y., Tani, J.: Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model. Neural Networks, 19, 323–337 (2006)
9. Jaeger, H., Haas, H.: Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. Science, 304(5667), 78–80, (2004)
10. Kandel, E. R., Schwartz, J. H., Jessell, T. M.: Principles of Neural Science. McGraw-Hill Medical, 4th ed., New York, USA (2000)
11. Maass, W., Natschläger, T., Markram, H.: Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. Computation, 14, 2531-2560 (2002)
12. Mandelbrot, B. B.: The Fractal Geometry of Nature. Freeman, Scan Francisco, USA, (1982)
13. Newman, M.E.J.: The Structure and Function of Complex Networks, SIAM Review 45, 167-256 (2003)
14. Pfeifer, R., Scheier, C.: Understanding Intelligence. MIT Press, Cambridge, USA (2000).
15. PlanetMath.org, `http://PlanetMath.org`
16. Wermter, S., Palm, G., Elshaw, M.: Biomimetic Neural Learning for Intelligent Robots. Springer, Heidelberg, Germany (2005)