A Connectionist Architecture for the Evolution of Rhythms

João M. Martins and Eduardo R. Miranda

Interdisciplinary Centre for Computer Music Research
University of Plymouth, Plymouth, Devon PL4 8AA
United Kingdom.
{joao.martins,eduardo.miranda}@plymouth.ac.uk

Abstract. In this paper we propose the use of an interactive multi-agent system for the study of rhythm evolution. The aim of the model proposed here is to show to what extent new rhythms emerge from both the interaction between autonomous agents, and self-organisation of internal rhythmic representations. The agents' architecture includes connectionist models to process rhythmic information, by extracting, representing and classifying their compositional patterns. The internal models of the agents are then explained and tested. This architecture was developed to explore the evolution of rhythms in a society of virtual agents based upon imitation games, inspired by research on Language evolution.

1 Introduction

The early applications of evolutionary computation to music go back to 1991 with the works of Horner and Goldberg by applying genetic algorithms to thematic bridging [1]. Since then there have been many successful attempts to apply these techniques to music. For a discussion on the history and achievements genetic algorithms please refer to Gartland-Jones and Copley [2].

Neural Networks have also been used extensively in the context of music. There have been connectionist models for pitch perception, rhythm and metre perception, melody conduction and composition, many of them collected in Griffith and Todd's book [3].

Memetic theory, the cultural counterpart of biological evolution, was invented by Dawkins in 1979 [4], and postulates that culture is an evolutionary process evolving through the exchange, mutation and recombination of units of information that can be observed in different scales. Although the definition of a meme is still quite obscure, there have been some computational attempts to model the evolution of musical style according to this theory [5].

In the specific case of rhythm composition, we can find applications of evolutionary computation such as the Interactive Genetic Algorithm (IGA)from Horowitz [6] to breed drum measure loops, the CONGA system from Tokui and Iba [7] using genetic algorithms and genetic programming to generate rhythms which are evaluated by the user, and the creation of rhythms with cellular automata by Brown [8].

All these methods have been developed mainly with three applications in mind: Sound synthesis, composition, and musicology [9]. This paper focuses on the later; i.e., a framework for the study the evolution of music.

2 Imitation Games: Language and Music

Agent based modelling is a technique frequently seen in the A-Life context to study complex systems. The emergent behaviour of the system is observed when autonomous elements self-organise as a consequence of the interactions between each other and the environment. Regarding music, the applications of A-life models are described by Miranda and Todd [10]. The scope of the work presented on this paper considers a society of agents where rhythms are exchanged, processed and categorised with neural networks.

In the real world, there is no direct transposition of the knowledge between individuals, this meaning that it is not possible to copy all the information inside a person's brain and present it to another. In the case of language or a musical performance, this features get more accentuated as there is a strangulation in the channel and consequently in the amount of information that you are able to process. Although, is easy to exchange information in the computer without loss of data, for the purpose of simulation we need to find processing mechanisms and interaction schemes that can cope with this human limitation.

While some defend the innateness of Language and thus the role of genetic mutations in its evolution, Steels [11] defends that language corresponds to a Self-organising phenomena like the ones observed in chemical and biological processes. Furthermore language develops subject to big pressures of the environment, such as limited time for articulation of words, and acoustically adverse environments.

The same duality of opinions can arise on the musical side. The transmission media is the same as language, and music is also subject to the same kind of pressures, although not constrained to meanings and concepts. Werner and Todd [12] put emphasis on the role of mate selecting pressures for the evolution of repertoires, and the evaluation of the specimen fitness is made the according to the musical material. Miranda [13] explored the self-organising potential of agents' societies by furnishing the agents with motor and auditory skills and letting them evolve a shared repertoire of short sound sequences through imitation games .

Originally inspired by Wittgenstein [14], Luc Steels [15] proposed a model of imitation games for artificial agents. Bart de Boer [16] applied this game methodology to study the emergence of a coherent vowel system handling phono-articulatory parameters. Miranda [17] applied a slightly different version of the algorithm to develop intonations. Basically the game consists of one agent picking a random sound from its repertoire and the other agent trying to imitate it. Then feedback is given about the success of the imitation. On the basis of this feedback, the agents update their vowel repertoires.

Our approach differs from the applications previously presented in the sense that the judgement is made upon a system of internal categories of each of the agents and how the repertoire evolves in the continuous search to generate music that the other agent will recognise in his internal categories system.

In this paper we introduce the groundwork that characterises our approach; i.e., the connectionist nature of the agent's mechanism for representing rhythms.

3 Agents Architecture

We will present the architecture an agent containing two neural networks in cascade that receive a stream of rhythmic events as input and contain three output neurons that map these rhythms into a tridimensional space. For a comprehensive foundation on neural network theory please refer to Haykin's book [18].

Each agent is provided with a set of two neural networks: a SARDNET and a one layer Perceptron (Figs 2 and 5). The first one receives the stimulus sequentially from an input, encoded as a MIDI stream of rhythmic events, and generates an activation pattern corresponding to the agents perception of the type of event and its place in the sequence. The dynamics of this network is fully explained in Sec. 3.1. The pattern of activation from the Sardnet then becomes the input of the later network, the Perceptron, which generates three output values that enable the categorisation of the received sequences. The architecture and learning rules of the Perceptron are explained in Sec. 3.2.

The events are represented as vectors with three components. The first component defines the musical instrument (timbre), the second defines the loudness (velocity), and the third defines the value in milliseconds that the sound lasts (Inter-onset interval). These three dimensions correspond to human perceptual attributes with different scales in sensitivity and range. Modelling these differences in the learning algorithm was not part of the scope of this paper.

3.1 Sardnet

The SARDNET [19] is a self-organising neural network for sequence classification that was applied in phonology and recently it was also applied to simulations for evolving melodies [20]. This network is an extension of the original Self Organised Map (SOM) which is a neural network used for unsupervised learning developed by Kohonen [21]. The SOM has proven to be a powerful tool for many engineering applications and some of its variations have provided explanations for the organisation and development of the visual cortex [22].

The SOM is also called a competitive network or "winner-takes-all" net, since the node with largest input "wins" all the activation, which reflects on the possibility of updating that unit in order to become more similar to the input. The neighbouring units of the winning neuron are also updated according to a neighbourhood function that organises representations of similar stimuli in a topographically close manner.

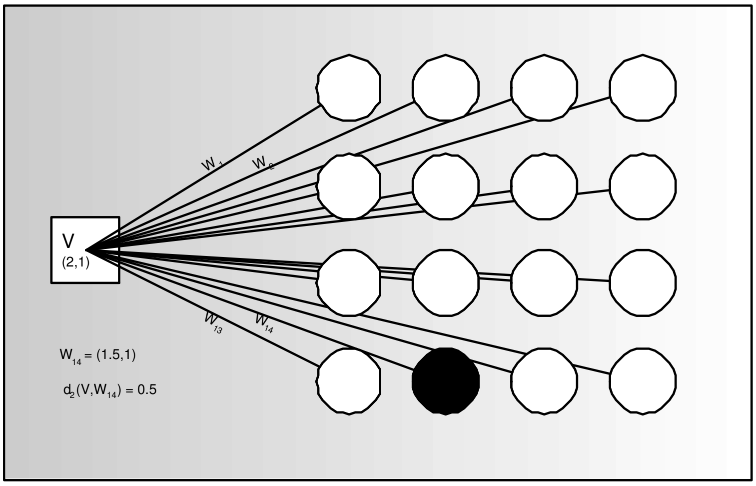

Fig. 1. Kohonen Self Organising Feature Map (SOM)

In Fig. 1 we can see a diagram of a SOM with 16 units and one input. The dimension of the input vector determines the dimension of the weights vector of each unit. To determine which weight vector is the closest one to the input unit, the euclidean distance is calculated:

$$d_2(\mathbf{v}, \mathbf{w}) = \sqrt{\sum_{i=1}^{n} |v_i - w_i|^2} \tag{1}$$

The SARDNET keeps some essential features from the SOM, but adds two important features that enables us to deal with sequences of events. The first diverging characteristic is that the winning neuron is removed from subsequent competitions, and the second difference corresponds to holding the previous activations with a decay in each time step. The dynamics of SARDNET is shown on Fig. 2 where we can observe a the stream of events passing through the input and activating three units in sequence (W_{14}, W_7, W_2). The training algorithm for the SARDNET is shown on Tab. 1.

Like the SOM, the SARDNET uses the Euclidean distance $d_2(w, v)$ from Eq. 1 to evaluate which is the weight that better matches the input. On step 3 of the algorithm the weight of the winning and the neighbourhood units are changed according to the standard rule of adaptation:

$$\Delta w_{jk} = \alpha(w_{jk,i} - v_i) \tag{2}$$

where α depends also on the distance to the winning unit, meaning its position in the neighbourhood. The neighbourhood function decreases as the map becomes more organised.

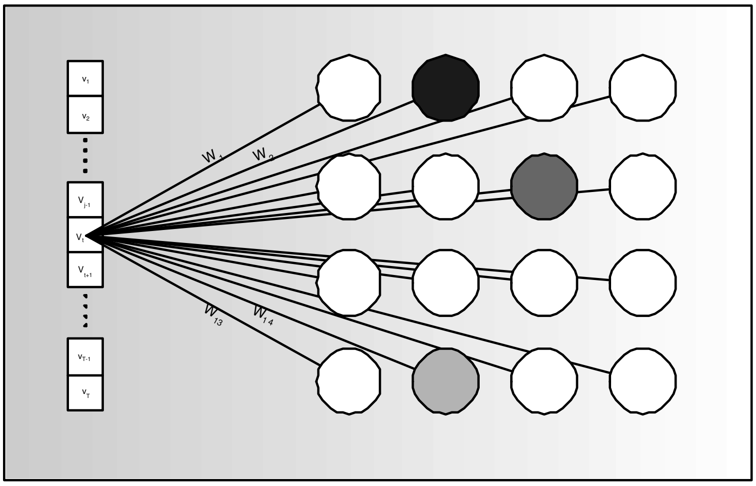

Fig. 2. SARDNET - Self-organising activation, retention and decay network

As in step 5 of the algorithm, all the active units are decayed proportionally to the decay parameter d,

$$\eta_{jk}(t+1) = d\eta_{jk}(t), \qquad 0 < d < 1 \tag{3}$$

	0	1	2	3	4
0	0.9	.	.	.	.
1	.	.	.	.	.
2	.	.	.	.	.
3	.	.	.	0.66	.
4	.	0.73	.	.	0.81

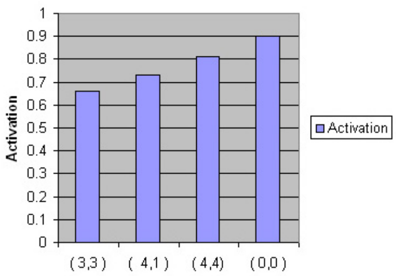

Fig. 3. Activations from the output layer on in two different views

In the following section we present the details of the Perceptron, the network that receives the activation patterns from the SARDNET, keeping the relevant information about this activation patterns across several sequences.

INITIALIZATION: Clear all map nodes to zero
MAIN LOOP: While not end of sequence 1. Find inactive weight vector that best matches the input. 2. Assign 1.0 activation to that unit. 3. Adjust weight vectors of the nodes in the neighbourhood. 4. Exclude the winning unit from subsequent competitions. 5. Decrement activation values for all other active nodes.
RESULT: Sequence representation = activated nodes ordered by activation values.

Table 1. The Sardnet training algorithm

3.2 Perceptron

The Perceptron is a neuron-like learning network developed by Rosenblatt [23] which is a one layer feed-forward neural network with a set of inputs that are fully connected to an output layer. The outputs of Perceptrons are explicit functions of the inputs. Fig. 5 shows its architecture.

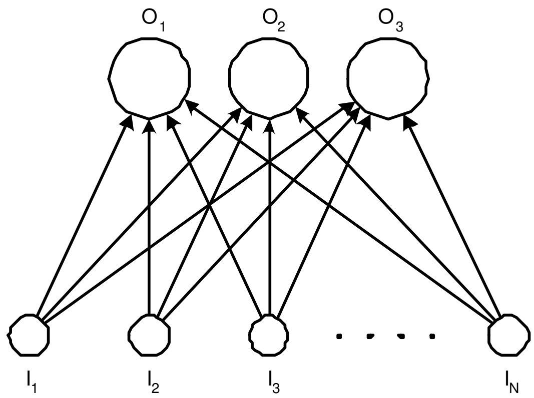

Fig. 4. Perceptron network

$$O_i = g(h_i) = g\left(\sum_k w'_{ik} I_k\right) \tag{4}$$

Eq. 4 is the propagation function of the Perceptron and $g(h)$ in Eq. 5 is the activation function computed by the units. In this case this function is a sigmoidal function,

$$O_i = g(h_i) = \frac{1}{1 + \exp(-h_i)} \tag{5}$$

The Perceptron uses the gradient descendant method to change the weights in order to adjust the test input to a given target.

$$\Delta w_{jk} = \eta * (T_k - O_k) I_j; \tag{6}$$

where η is the learning rate, T is the target value and $T_k - O_k$ is the corresponding error during the training phase.

The number of inputs of the Perceptron is the number of units of the SARDNET. The number of output neurons is arbitrarily defined as being 3 to be able to visualise the results in a tridimensional grid. This output grid enables the categorisation of the input sequences.

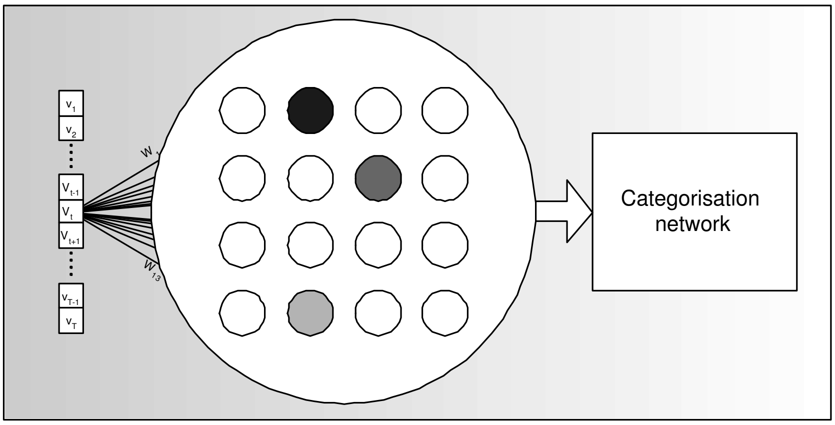

Fig. 5. Interaction diagram for the imitation game proposed

4 Analysis of the Agent

4.1 Sardnet

First we trained the Sardnet solely with prerecorded rhythms. We used a map with 50 elements, 10 in the length and 5 in the breadth, a learning rate of 0.1. The map was initialised with random weights in the range of -1 to 1. To perform the first organisation tasks the map was fed with 5 sequences of rhythms of latin music, each of them containing one or two instruments, very much like it would be if these were performed by other agents. After a couple of iterations a pattern of organisation could already be observed in the network, but the correspondent sequences extracted sounded extremely chaotic. After 50 iterations the rhythms start to sound organised as well, and the changes to the timbre of the instrument have the largest perceptual impact. This was expected to be so, as there is no discrimination in the organisation algorithm regarding the different weight components. Nevertheless, the organisation process is fine tuned enough to adapt perceptually perfectly to the incoming sequence after 80 iterations, and a learning musician is also expected to make timbre mistakes.

The graphs from Fig. 6 show the evolution of the third component of the weights (Inter-onset Intervals). The first graph shows the initial value of the weights, as explained above, the second shows the organisation process after 20 iterations, and the third shows the weights stabilised after 80 iterations. Fig. 6 d) shows the difference between the sums of the weights in two consecutive iterations, this being a measure of the stabilisation of the weights.

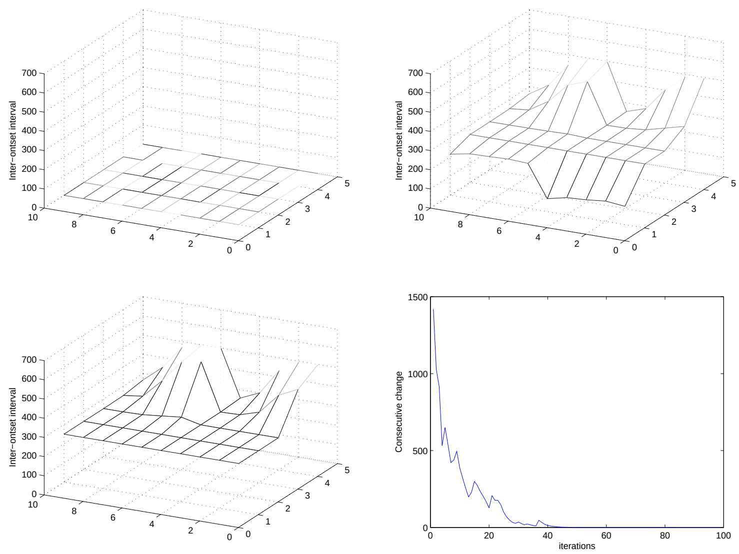

Fig. 6. Sarnet weight evolution without change of neighbourhood: a) Weight initialisation; b) After 20 iterations; c)After 80 iterations; d) Difference between the weights' sum in consecutive iterations

Previously it was stated that the SOM adapts its weights, not only for the winning elements, but also in its neighbourhood. In Fig. 7 it is shown the same organisation process but considering the neighbourhood change. The parameter σ controls the range of the the gaussian that changes the neighbourhood. By using an initial value of $\sigma = 2.97$ we can more rapidly capture the global characteristics of the input. It is necessary to reduce gradually this value in order not to destroy the representations of the events that occur less frequently. Comparing Figs. 6d) and 7d) we see that this procedure accelerates the convergence process.

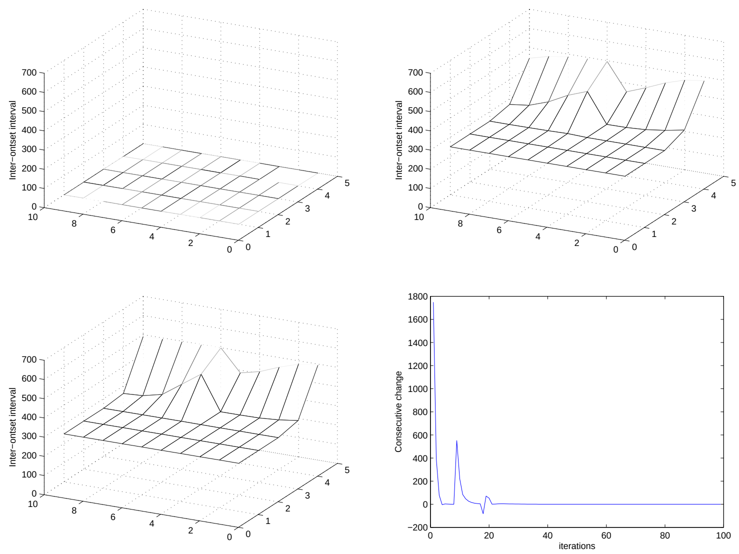

Fig. 7. Sarnet weight evolution with change of neighbourhood: a) Weight initialisation; b) After 20 iterations; c)After 80 iterations; d) Difference between the weights' sum in consecutive iterations

One of the most important conclusions is that although it is possible to extract very similar sequences from both maps, the internal representation can be quite different, as can be seen from both Figs. 6 and 7 both trained with the same sequences.

4.2 Perceptron

The Perceptron's architecture is explained in Sec. 3.2. The Perceptron used for these experiments had 50 input units, that receive their values directly from the activations of the output layer of the Sardnet. These input units are fully connected to 3 output neurons enabling the mapping and categorisation of the input sequences into a tridimensional space of straightforward visualisation. We chose the first three activation layers of 50 elements corresponding to three rhythms fed previously to the Sardnet, and trained the Perceptron to respond to these patterns with three different targets, namely $[1, 0, 0]$, $[0, 1, 0]$ $[0, 0, 1]$. This process took 434 epochs to reach an error of categorisation of 10^{-3} as can be seen in Fig. 8 a). Each training patterns is marked with an (o) in the categorisation space (Fig. 8 b)). Later, we fed the perceptron with the last two rhythms and observed its activation marked with an (x). These were found to be much closer to the $[0, 1, 0]$ target, which interestingly correspond to the most similar pattern regarding the IOIs.

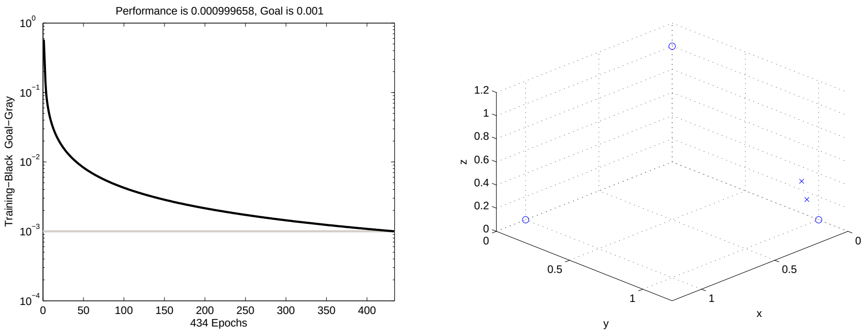

Fig. 8. a) Perceptron error in learning process; b) Categorisation space

5 Conclusion

With this paper we presented the architecture of an interactive virtual agent that is able to learn rhythms. The agent is composed of two neural networks that are able to learn the rhythms representation through self-organising processes. As it happens with humans, the agents always have different internal representations for the rhythms they listen to. Furthermore, the output of the networks categorises the incoming sequences and provides a measurement for the agents to judge how related are the listened rhythms. The rhythm representation allows for all types of rhythms to be encoded, considering event variables of Inter-onset interval, timbre and intensity. Several tests to the individual networks were made to show the potential to evolving rhythms and categories. We are now studying the results of number of simulations of imitation games where different rhythmic repertoires were evolved from scratch under a variety of different scenarios.

References

1. Horner, A., Goldberg, D.: Genetic algorithms and computer-assisted music composition. In: Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, CA, Morgan Kauffman (1991)
2. Gratland-Jones, A., Copley, P.: The suitability of genetic algorithms for musical composition. Contemporary Music Review **22**(3) (2003) 43–55
3. Griffith, N., Todd, P.: Musical Networks. MIT-Press, Cambridge:USA (1999)
4. Blackmore, S.: The Meme Machine. Oxford University Press (1999)
5. Gimenes, M., Miranda, E.R., Johnson, C.: A memetic approach to the evolution of rhythms in a society of software agents. In: Proceedings of the 10th Brazilian Symposium of Musical Computation (SBCM), Belo Horizonte (Brazil) (2005)
6. Horowitz, D.: Generating rhythms with genetic algorithms. In Anderson, P., Warwick, K., eds.: Proceedings of the International Computer Music Conference, Aarhus(Denmark), International Computer Music Association (1994)
7. Tokui, N., Iba, H.: Music composition with interactive evolutionary computation. In: Proc. 3rd International Conf. on Generative Art, Milan, Italy (2000)

8. Brown, A.R.: Exploring rhythmic automata. In Rothlauf, F., et al., eds.: Proceedings of the 3rd European Workshop on Evolutionary Music and Art, Lausanne(Swizerland), Springer Verlag (2005)
9. Coutinho, E., Gimenes, M., Martins, J., Miranda, E.R.: Computational musicology: An artificial life approach. In: Proceedings of the 2nd Portuguese Workshop on Artificial Life and Evolutionary Algorithms Workshop, Covilhã(Portugal), Springer Verlag (2005)
10. Miranda, E., Todd, P.: A-life and musical composition: A brief survey. In: Proceedings of the IX Brazilian Symposium on Computer Music, Campinas,(Brazil) (2003)
11. Steels, L.: The synthetic modeling of language origins. Evolution of Communication **1**(1) (1997) 1–34
12. Werner, G., Todd, P.: Too many love songs: Sexual selection and the evolution of communication. In Husbands, P., Harvey, I., eds.: ECAL97, Cambridge, MA, MIT Press (1997) 434–443
13. Miranda, E.R.: Emergent sound repertoires in virtual societies. Computer Music Journal (MIT Press) **26**(2) (2002) 77–90
14. Wittgenstein, L.: Philosophical Investigations. Blackwell Publishers (1979)
15. Steels, L.: A self-organizing spatial vocabulary. Artificial Life **2**(3) (1995) 319–332
16. de Boer, B.: Self-Organisation in Vowel Systems. PhD thesis, Vrije Universiteit Brussel AI-lab (1999)
17. Miranda, E.R.: Mimetic development of intonation. In: Proceedings of the 2nd International Conference on Music and Artificial Intelligence (ICMAI 2002), Springer Verlag - Lecture Notes on Artificial Intelligence (2002)
18. Haykin, S.: Neural Networks. Prentice Hall, New Jersey:USA (1999)
19. James, D.L., Miikkulainen, R.: SARDNET: a self-organizing feature map for sequences. In Tesauro, G., Touretzky, D., Leen, T., eds.: Advances in Neural Information Processing Systems 7, Cambridge, MA, USA, MIT Press (1995) 577–84
20. Bosma, M.: Musicology in a virtual world: A bottom up approach to the study of musical evolution. Master's thesis, University of Groningen (2005)
21. Kohonen, T.: Self-Organizing Maps. Springer Series in Information Sciences. Springer-Verlag Berlin and Heidelberg GmbH & Co. K (1997)
22. Bednar, J.A., Miikkulainen, R.: Joint maps for orientation, eye, and direction preference in a self-organizing model of V1. Neurocomputing (in press) (2006)
23. Rosenblatt, F.: Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. Spartan Books, Washington (1962)