Playing Integrated Music Knowledges with Artifical Neural Networks

Frédéric VOISIN and Robin MEIER
CIRM, 33 Avenue Jean Medecin, Nice; May 2004

Abstract

This text presents some aspects of the Neuromuse[1] project developed at CIRM. We will present some experiences in using self-organizing maps (SOM)[2] to generate meaningful musical sequences, in real-time interfaces, with MaxMSP and Jitter[3].

1 Introduction

Recent advances in the last decades in cognitive sciences, neurobiology and computer sciences make artificial neural networks (ANN) more pliable in musical applications. This connectionist paradigm, which is dramatically different to traditional computing, requires a variety of approaches, from the more factual (biology, physics, computing) to the more abstract (sciences of cognition, sociology, artificial intelligence).

2 SOM in MaxMSP

jit.robosom is the first freely distributed self-organizing map for MaxMSP-Jitter, written at CIRM by Robin Meier[4]. A SOM is normally used to represent and extract meaningful content from raw data. It can be applied to classification tasks, pattern recognition, filtering, and data completion. The ability to use a SOM within a real time musical environment simplifies a number of tasks: it's easy to work with audio, symbolic streams and logics. Raw-data (i.e. stimuli) can be encoded and sent directly to a SOM within Max using Jitter matrices. Furthermore it is now easier to use a SOM in real time for high-level user control and augmented interfaces. The Jitter layer provides another way for encoding large vectors, and gives the user a graphical representation of the activation map of the neural network: neural networks are not black-boxes anymore, one can now see their full activity, just as a neurologist can visualize brain activity with medical imaging technologies. Furthermore, these activation maps of ANN can be recorded as movie files in sync with their input and output data for later observation and analysis. Using standard Max objects for programming artificial neural networks provides a convenient way of changing their code on the fly: the generic algorithm of a SOM is quite simple and can be edited using Max programming to adapt it to peculiar tasks or experiments. One can now design and experiment with complex networks using several content access memories, and feed-forward networks for data computing and encoding. Using various programming languages within Max or Pure-Data in real-time, such as LISP or Java, makes the use of distributed agent systems more flexible. One can also use genetic algorithms to generate learning rules and network architectures[5].

In the public distribution of *jit.robosom* two simple chaotic features are provided[6]. A temperature factor which is inspired by the simulated annealing method found in heuristic algorithms, and adds noise to the "synaptic" weights. Another temperature factor adds slight errors to the distance evaluation of the patterns, making classification more or less precise. Each of these variations in the code and learning rules may introduce differences in the ANN behaviors. Various chaotic or highly dynamic rules may be applied and studied[7].

[1] http://www.neuromuse.org

[2] Teuvo Kohonen : Self-Organizing Maps, Springer Series in Information Sciences, Vol. 30, 1995; 1997, 2001.

[3] http://www.cycling74.com

[4] http://www.neuromuse.org/downloads

[5] See Balakrishnan Honavar : Evolutionary Design of Neural Architectures, 1995, Iowa State University

[6] see Robin Meier, Applications musicales de cartes auto organisatrices: *jit.robosom*. http://robin.meier.free.fr/memoire.pdf

[7] Christophe Philemotte, "Etude des réseaux neu-

Changing the architecture of the ANN can be done in different ways, at different levels in the network. This can range from slight tunings to strong changes of the generic rules and network topology. Experimenting with pathologic or minimalistic ANN can help to understand their general behavior as much as successful ones. In music production these experiences constitute a highly interesting empirical material not only for the study of ANN, but also for the study of musical knowledge.

3 From classification to cognition

Classification is a standard application of self-organizing maps. High dimensional data can be reduced to a lower dimensional space. The relevancy of the classification is due to i) the encoding (pre-processing), ii) the topology, and iii) the network's process of learning. Preprocessing can be done both by distributed memories and feed-forward networks, or by classical algorithms. Figure 1 shows a basic example using *jit.robosom* in Max for vowel classification. Here, the preprocessing consists only in extracting the first 256 bins of a 512 bins FFT, computed directly from the sound samples (vowels).

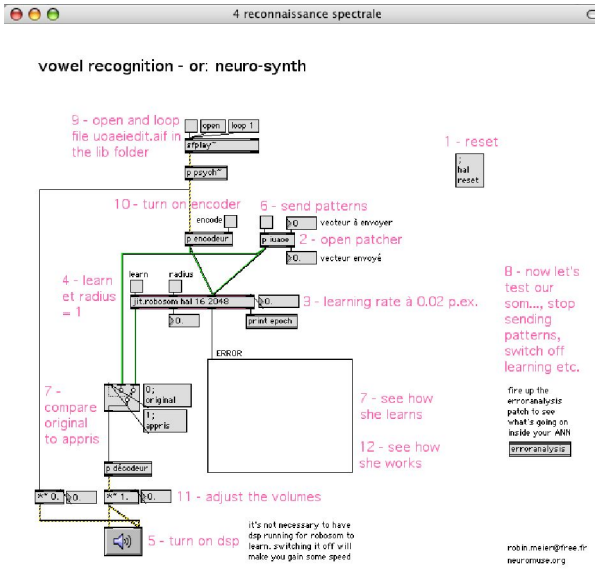

Figure 1: Example-patch for *jit.robosom*

ronaux en tant que systèmes dynamiques chaotiques", http://iridia.ulb.ac.be/~cphilemo/tfe/tfe.html and: Emmanuel Daucé, "Adaptation dynamique et apprentissage dans de réseaux de neurones récurrents aléatoires", http://esm2.imt-mrs.fr/~dauce/these/

The activation map of the SOM in Jitter gives the details of the classification during learning and processing in an arbitrary space chosen by the user (see figure 2). In this example, a 16 unit SOM makes a square map in which four vowels may be classified. The observation and the analysis of the activation during learning can give an abundance of relevant information about the processed data. The history of SOM activations and errors in the learning process is part of the data analysis, showing how a classification is processed within an arbitrary context. As the video HAL04.mpg[8] demonstrates in figure 2, the well-known learning process of a SOM reveals phonological aspects of the stimuli through the activations of its 16 units. This also demonstrates how similarity is not a stationary category, but can refer to dynamic and temporal aspects defining a context. Since a SOM learns and processes solely by impression and imitation, it is simply processing a stimulus through the image of its history that has been collapsed into one frame (as well as some recurrent connections). An example of such an actual activation is shown in figure 2, where the image represents the activation map of the 16 units-SOM, as a memory recall of the stimulus during the learning process. The sound resynthesis is performed by a FFT-1 from these activations.

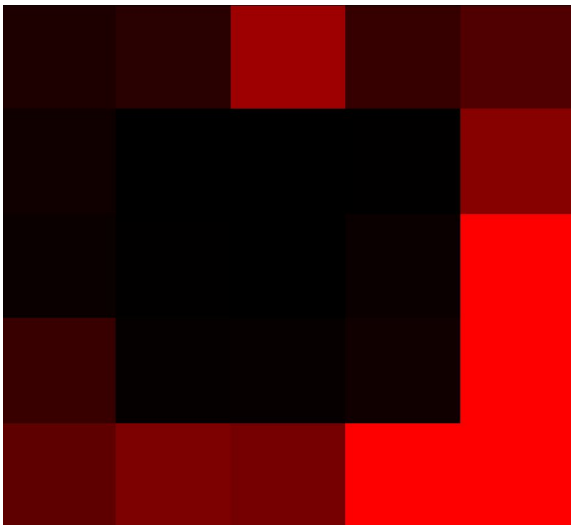

Figure 2: activation map of a 16 units SOM with Max-Jitter

The observation of this example clearly shows that the functional area of the network corresponds to implicit small units of knowledge (phonems) which are transmitted via ordered sets of stimuli,

[8] http://www.neuromuse.org/txt/tutorial/Hal04.html

making more or less explicit relations. It also shows that the control of stimuli and learning rules gives full control of a SOM behavior: if new stimuli are presented to a SOM following any arbitrary path, the SOM will classify first, according to its initial state and then it adapts, if necessary, to these new aspects. The study of these relationships between knowledge representations and memory topologies, is an interesting approach for generating new and original musical material with a SOM.

4 Playing with knowledge units

The following example is a program in Max using two small-sized SOM (26 memory units, "neurons") generating musical sequences in MIDI format, according to given examples. Short patterns of 15 to 30 notes encoded in a 3D-space (pitch, velocity, inter-onset interval) are played to a single unit SOM doing a time loop processing of the input with feedback. Temperature and sensitivity (learning factor) of the SOM give a control of looping variations of the musical sequence, making, in a minimalist way (one unit-automat), deviations from original patterns (figure 3).

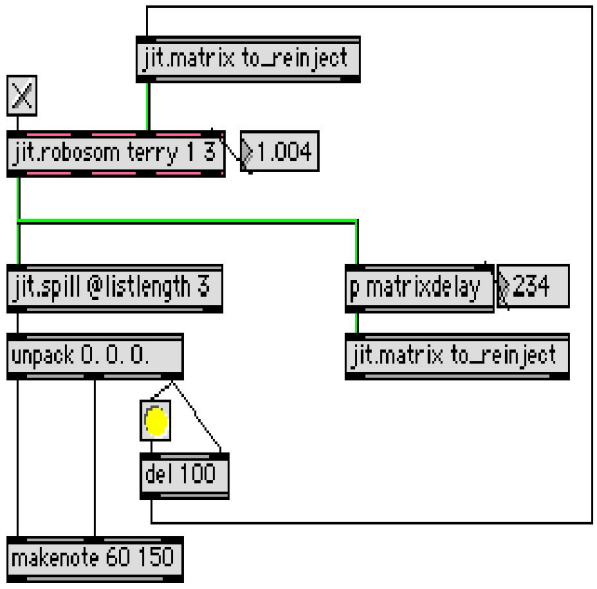

Figure 3: single unit SOM doing a time loop process

At the same time, this looped-with-variation output is sent to a 25 unit SOM trained to repeat the stimuli, according to its actual state (see example figure 4), which depends on what it has recently learned. Temperatures and sensitivity of the SOM are particularly interesting parameters to control in real-time. They permit numerous methods of variation, from perfect repetition to meaningful interpolations, "spontaneous" generation of new "relevant"material or chaotic cycles, in relation to the given and ordered stimuli. In this example, the 25 unit SOM outputs new patterns related to previous (already "heard") and actual ("unknown") patterns. It's adapting itself to the new (melodic) context coming from the single unit looping SOM, or from the user (a musician playing with Max interface). It shows how a SOM can react as a knowing agent, receiving new stimuli, memorizing and classifying them according to the implicit rules it was just exposed to. Temperature control and other dynamic features may help a SOM jump to new types of representations, producing new and relevant variations.

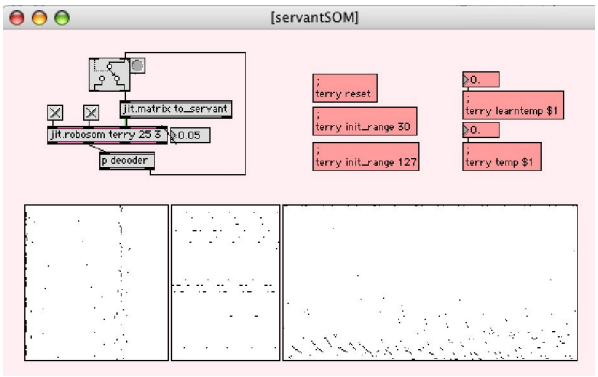

Figure 4: 25-units SOM trained to repeat the stimuli

As soon as we are interested in these dynamic processes of the network, the traditional fitness error, at some stage, may not be a good criterion for evaluating a neural network's capabilities. In such a context, the output of a trained SOM with large fitness errors is not hazardous (as is the case when using an untrained SOM). Some aspects of this empirical material may not only question the connectionist approach involved; it also questions original human representations of musical knowledge. Then, one can experiment with the hypothesis in which so-called natural knowledges and languages can adapt themselves to neurophysiologic properties of the brain[9]: computational conflicts, errors, and mismatches can reveal conflicts at a higher cognitive level that can be advantageously used for gen-

[9]Cf. Simon Kurby, Morten Christiansen, NewScientist,18/01/03, p. 30

erating original musical material.

As shown with the following musical example for sampled piano and *jit.robosom* ("For Alan Turing", by Robin Meier, 10'30), the minimalistic self-organizing maps are not able to integrate the entire musical material presented to them. Therefore they are becoming dynamic musical constraints, based on a perception of constantly changing musical events. Playing with this perception in real-time through the learning process and the maps predictibility permits an interaction with these dynamic properties. Later developments involving more sophisticated architectures and more complex musical structures remain to be studied.

5 Conclusion

This approach shows the contemporary ability for a large community of musicians to easily perform cognitive processing to generate music with artificial neural networks. The understanding of the complex behavior of ANN needs much investigation and experimentation for their use in musical applications. Experimenting with such knowledge-based systems is not only valuable from a scientific perspective, but also a source of various musical inspirations.