Self-Organization of Feature Detectors in Time Sequences (SOFT) — A Neural Network Approach to Multidimensional Signal Analysis

Axel Wismüller
Institut für Radiologische Diagnostik (Director: Klaus Hahn)
Ludwig-Maximilians-Universität München, Klinikum Innenstadt
Ziemssenstr. 1, D-80336 München, Germany
e-mail: Axel.Wismueller@physik.uni-muenchen.de

Herbert Jaeger
GMD
FIT.KI
Sankt Augustin, Germany

Dominik R. Dersch
Dept. of Electrical Engineering
University of Sydney
Australia

Helge Ritter
AG Neuroinformatik
Universität Bielefeld
Germany

Günther Palm
Abt. Neuroinformatik
Universität Ulm
Germany

Abstract

In this paper, we present a neural network algorithm for self-organization of feature detectors in time sequences (SOFT) based on the mathematical concept of transient attractors. It evaluates local phase space volume contraction as an indicator for good short-term predictability. SOFT supports category formation and event detection in multidimensional time sequences by linking together neural function approximation and principal component analysis. Possible extentions of the algorithm including iteration and vector quantization procedures for further data analysis are discussed.

1 The concept of transient attractors

What enables biological systems to recognize discrete events in a continuous stream of sensory input data? How can they learn to detect and categorize elementary feature patterns in continuous time sequences without explicit knowledge of such re-recognizable entities? How can psychologists and ethologists identify elementary behaviors when observing continuous motion patterns produced by humans, animals or robots? All these problems imply a high-dimensional continuous dynamics which gives rise to a sequence of discrete nameable and re-recognizable regularities. It is of obvious importance for many fields of science including biomedical research, economics, cognitive science, psychology, computer vision, automatic speech recognition, robotics etc. to develop mathematical models of such identifiable events and corresponding algorithms which enable us to extract them from empirical time sequence data — in other words, techniques for transforming non-symbolic time series into discrete symbol sequences.

In this paper, we present a neural network algorithm for self-organization of feature detectors in time sequences based on the mathematical concept of *transient attractors*. The key idea is to identify re-recognizable regularities in time sequences by good short-term predictability implying local contraction of phase space volume. In contrast to former approaches based on the analysis of raw data, we employ neural network models, i.e. *parametric representations* of data sets in order to cope with the sparseness of time sequence trajectories in empirical high-dimensional phase-portraits. At the same time, we avoid the computational burden involved by a complete search of the data set for investigating the neighborhood of each trajectory point. SOFT combines neural function approximation, principal component analysis, and vector quantization within a combined computational procedure.

0-7803-4859-1/98 $10.00©1998 IEEE

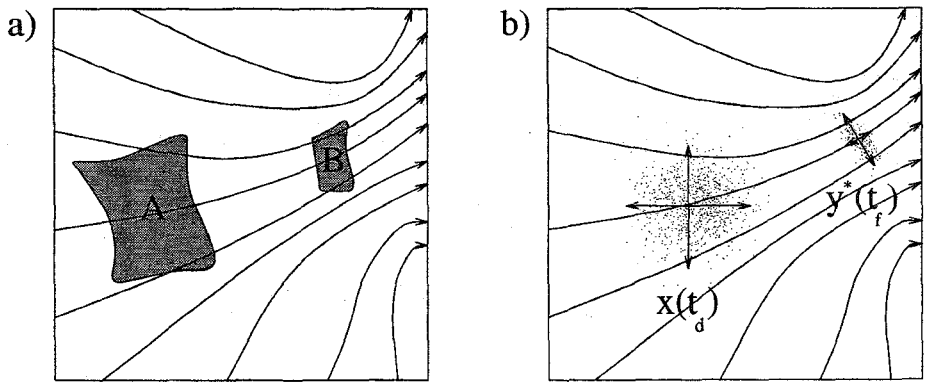

Figure 1: The SOFT algorithm: intuitions. a) contraction of phase space volume in regions of converging trajectories. b) the SOFT principle: The empirical trajectory point $\mathbf{x}(t_d)$ is "blurred" by superimposed noise. The resulting source distribution forms the input of a neural network function approximator predicting a target distribution in the neighborhood of the empirical trajectory point $\mathbf{y}^*(t_f)$. Transient attractors are identified by comparative evaluation of source and target distribution employing principal component analysis and calculation of local Liapunov numbers, see text.

How can we detect "nameable regularities" in time sequences? Two different families of algorithms exist for this task. The first approach relies on *partitioning* the phase space of an observed system into suitable cells, which can be labelled by symbols. As the system trajectory passes through these cells, a series of symbols is derived in a natural way. This kind of approach is standard in many fields, like in ergodic theory, in the analysis of the dynamics of recurrent neural networks [4], or the theory of qualitative reasoning in artificial intelligence, e.g. [10]. The second approach is based on *attractors* which are identified with re-recognizable elementary events. This perspective is popular in some strands of brain research (e.g. [16]) or connectionism (e.g. [13]). A survey on both schools can be found in [8].

The two approaches have complementary merits. Attractors are inherently stable. A drawback of attractors is that, strictly speaking, a dynamical system cannot leave an attractor state. Therefore, additonal mechanisms have to be included into formal models in order to account for the fact that a *sequence* of "attractor events" is produced by the system. Several such additional mechanisms have been explored in the literature. We refer to [7] and [9] for a review of this topic. Partition cells, conversely, lack any aspect of stability, but naturally give rise to sequences of symbolic units: simply collect the sequence of partition cells passed by the system trajectory.

This situation calls for an effort to combine both kinds of approaches into a unified mathematical construct of "regularities" or "events" which preserves the complementary merits of each of the original concepts discussed in the literature (see e.g. [8]). This paper is based on the construct of *transient attractors* [7], [9] which realizes such a combination.

The basic intuitive motivation of this concept can be summarized as follows: A first step towards "understanding" continuous, possibly high-dimensional, and possibly noisy time sequences is to look for some kind of "regularities" in the data. The idea of "regularity" is imprecise and has many facets. One of them is *short-term predictability*. Naively, short-term predictability means the following. As a human observes an empirical process (e.g. the behavior of an ant watched by an ethologist or some spiking neurons investigated by a neurobiologist), he will soon become aware that certain activity patterns occur repeatedly (e.g. the ant drops a pup at places where other pups lie, or some neuron A generates a spike burst whenever another neuron B stops firing). Repeated observations, in turn, enable the observer to make short-term predictions (e.g. the ant will drop the pup in the next instant since it approaches a heap of other pups; or neuron A will immediately begin to fire since the firing rate of B is dropping). In fact, it can be argued that short-term predictability is necessary for re-recognizability, i.e. for the very constitution of nameability.

The central idea within the concept of transient attractors is to use short-term predictability as the defining criterion for the identification of nameable regularities. A natural way for obtaining a precise concept of short-term predictability is to identify it with local contraction of phase space volume. This

corresponds to the standard view in information theory and ergodic theory (e.g. [11]). To get a graphical impression of local phase space contraction, assume that the empirical time series is plotted in a phase diagram (Fig. 1a). Then look for local regions where trajectories "converge". In such regions, called "transient attractors", the phase space becomes contracted in the sense that if the process is known to be in a volume element A at time t, then it can be claimed to be in a *smaller* volume element B at time $t+\Delta t$. This is a reduction of information-theoretic uncertainty, or "good predictability".

The intuition described so far can be cast into a precise mathematical definition of transient attractors. For a rigorous mathematical formalization of this concept and a detailed discussion of problems and pitfalls, we refer to the work of Jaeger [7], [9].

2 The SOFT algorithm

In order to identify transient attractors, we propose a sequence of computational procedures:

(i) Consider a K-dimensional time sequence $Z = \{\mathbf{z}(t)\}$, $\mathbf{z}(t) \in \mathbb{R}^K$, $K \in \mathbb{N}$. At any given time $t \in \{1,\ldots,T\}$, a D-dimensional feature vector $\mathbf{x}(t) \in \mathbb{R}^D$, $D \in \mathbb{N}$ and another F-dimensional feature vector $\mathbf{y}^*(t) \in \mathbb{R}^F$, $F \in \mathbb{N}$ may be extracted from Z within problem-specific preprocessing procedures. $\mathbf{x}(t)$ and $\mathbf{y}^*(t)$ describe empirical trajectories in phase spaces with dimensions D and F, respectively. For example, $\mathbf{x}(t)$ may contain information extracted from a window covering several adjacent frames of Z, in analogy to the input structure used in time-delay neural networks. The simplest case would be $D = F$, $\mathbf{x}(t) = \mathbf{y}^*(t)$ for all $t \in \{1,\ldots,T\}$ (as, for instance, shown in Fig. 1).

(ii) Now a neural network function approximator is trained to represent a *pre- or postdiction mapping*

$$f^* : \mathbb{R}^D \to \mathbb{R}^F,\ \mathbf{x}(t_d) \mapsto \mathbf{y}^*(t_f),$$

where $t_f = t_d + \tau$ with $t_d, t_f \in \{1,\ldots,T\}$ and a given $\tau \in \mathbb{Z}$ as a problem-specific pre- or postdiction interval.

We do not explicitly specify the neural network architecture of the function approximator: for example, implementational alternatives could be multi-layer-perceptrons trained by the error-back-propagation algorithm [12] or generalized radial-basis-functions-networks (see e.g. [5], [2], [14]).

(iii) In the following, $g(\mathbf{x};\mu,\Sigma,B)$ denotes a B-dimensional Gaussian distribution of vectors $\mathbf{x}$ with mean μ and covariance matrix Σ, where $\mathbf{x},\mu \in \mathbb{R}^B$, $\Sigma \in \mathbb{R}^{B\times B}$, $B \in \mathbb{N}$, i.e.

$$g(\mathbf{x};\mu,\Sigma,B) := \frac{1}{(2\pi)^{B/2}(\det\Sigma)^{1/2}} \exp(-\tfrac{1}{2}(\mathbf{x}-\mu)^t\Sigma^{-1}(\mathbf{x}-\mu)). \quad (1)$$

Each data vector $\mathbf{x}(t_d)$ of the trajectory in (i) is "blurred" by superimposed Gaussian noise, i.e. data vectors $\mathbf{x}^\nu(t_d)$ are generated in the neighborhood of $\mathbf{x}(t_d)$ according to a *source distribution*

$$p(\mathbf{x}^\nu(t_d)) = g(\mathbf{x}^\nu(t_d);\mathbf{x}(t_d),\Sigma_\mathbf{x},D) \quad (2)$$

where, for simplicity, we restrict to a D-dimensional univariate Gaussian probability density, i.e.

$$\Sigma_\mathbf{x} = \delta^2\mathbf{1_D}$$

with $\delta > 0$ representing the scale of resolution for transient attractors to be detected and $\mathbf{1_D}$ the D-dimensional identity. This source distribution is mapped onto a *target distribution* by the neural network function approximator representing $f(\mathbf{x}^\nu(t_d)) = \mathbf{y}^\nu(t_f)$ (Fig. 1b). For small δ, f can be considered locally linear employing Taylor series expansion. As Gaussian distributions remain Gaussian under linear transformations of the random variable (for proof see e.g. [3]), the target distribution of the $\mathbf{y}^\nu(t_f)$ is approximately Gaussian:

$$p(\mathbf{y}^\nu(t_f)) = g(\mathbf{y}^\nu(t_f);\mathbf{y}(t_f),\Sigma_\mathbf{y},F), \quad (3)$$

where $\Sigma_\mathbf{y}$, in general, is no diagonal matrix.

(iv) *Principal component analysis* (PCA) of the target distribution reveals eigenvectors $\mathbf{u}_i$ and eigenvalues ϵ_i^2, $i \in \{1,\ldots,F\}$ of the covariance matrix $\Sigma_\mathbf{y}$ according to $\Sigma_\mathbf{y} = \mathbf{U}\Sigma_\mathbf{y}'\mathbf{U}^t$ with $\mathbf{U} = (\mathbf{u}_1,\ldots,\mathbf{u}_F)$ eigenvectors of $\Sigma_\mathbf{y}$, and $\Sigma_\mathbf{y}' = \mathrm{diag}(\epsilon_1^2,\ldots,\epsilon_F^2)$ representing the variances along the principal components of the target distribution.

(v) Now we could easily define a kind of "volume measure" for the encountered distributions: If we take $V_\mathbf{x} = \delta^D$ as the "volume" of the D-dimensional univariate Gaussian source distribution and $V_\mathbf{y} = \prod_{i=1}^F \epsilon_i$ as the "volume" of the target distribution, we could define the *contraction index* $c^T(t_f) = V_\mathbf{y}/V_\mathbf{x}$ as an indicator for

trajectory "convergence". The volume measure of the target distribution of the $\mathbf{y}^{\nu}(t_f)$ is not affected by the PCA in step (iv), as it is invariant under the resulting unitary transformation $\mathbf{y}^{\nu\prime}(t_f) = \mathbf{U}^t\mathbf{y}^{\nu}(t_f)$.

However, there may be situations in which convergence of empirical trajectories will not occur in all phase space dimensions, but will be restricted to certain subspace manifolds, whereas trajectories in other directions are not involved or even diverge (see Fig. 1b). In these situations, we may have to take direction-specificity of trajectory convergence into account. For this purpose, we propose a measure motivated by stability theory of dynamical systems: If we order the principal components $\epsilon_i, i \in \{1,\ldots,F\}$ of the target distribution from largest to smallest and let δ^2 denote the variance of the source distribution (see (iii)), we can calculate F numbers

$$\lambda_i := \ln(\frac{\epsilon_i}{\delta}). \qquad (4)$$

We call them local Liapunov numbers (LLNs). By calculating LLNs, we can investigate local phase space contraction by comparative evaluation of source and target distributions. Although our concept of LLNs is obviously motivated by a close analogy to the definition of Liapunov numbers in dynamical systems theory (see e.g. [6]), we want to emphasize three important differences:

a) LLNs do *not* represent a property of the *whole* trajectory, as we are not interested in the behavior of a given phase space volume as time approaches infinity. Rather do we focus on the local evolution of trajectories between finite times t_d and t_f separated by a given pre- or postdiction interval τ. This refers to the evanescent character of transient attractors defined by local two-point predictability.

b) LLNs are *not* restricted to the temporal evolution of trajectories in *one* given phase space. They are more general in a specific sense: Source and target distributions may be defined in different phase spaces according to possibly different procedures of feature extraction (see (i)). They are only linked together by the prediction mapping f of the neural network function approximator.

c) For the analysis of stability properties in dynamical systems theory, one is usually interested in the computation of the largest Liapunov numbers. In our context, however, we do not investigate long-term stability of a dynamical system, but focus our attention towards the smallest LLNs, as they represent local phase space contraction according to the definition of transient attractors.

By computing the LLNs λ_i, we can evaluate theoretical predictability *(T-predictability)* by analyzing local temporal phase space contraction with respect to direction-specific trajectory evolution. As a quantitative measure, we can obtain a contraction index $c^T(t_f)$ by calculating an appropriate function of the λ_i's. For example, $c^T(t_f) := \min_{i\in\{1,\ldots,F\}} \lambda_i$ would be a simple choice.

(vi) The predicted value $f(\mathbf{x}(t_d)) = \mathbf{y}(t_f)$ may differ from the real trajectory point $f^*(\mathbf{x}(t_d)) = \mathbf{y}^*(t_f)$ due to noise or to inevitable inaccuracies based on limited prediction quality of the neural network function approximator. For any pair $(\mathbf{x}(t_d), \mathbf{y}^*(t_f))$ of trajectory points, a decision has to be made whether $\mathbf{y}^*(t_f)$ belongs to a predicted target distribution. As a possible quantitative measure $c^P(t_f)$ for this practical predictability *(P-predictability)*, we propose a monotonuously decreasing function of the likelihood $L(t_f)$ of $\mathbf{y}^*(t_f)$ with respect to the parametrized target distribution, i.e.

$$L(t_f) = g(\mathbf{y}^*(t_f); \mathbf{y}(t_f), \mathbf{\Sigma}_\mathbf{y}, F) \qquad (5)$$

which can be calculated on the basis of steps (iii) and (iv) without additional computational expense.

(vii) Coupling the aspects of T- and P-predictability by an appropriate heuristic combination of $c^T(t_f)$ and $c^P(t_f)$, we can obtain a quantitative measure $c(t_f)$ for the actual local predictability at any given time t_f. We can identify trajectory points with a low $c(t_f)$ as indicators for the presence of a transient attractor.

3 Extentions and points of discussion

How many data points $\mathbf{x}^{\nu}(t_d)$ should be used in step (iii)? At first glance, the complexity of the algorithm is $O(F^2)$ in this context, for there are $\frac{F(F+3)}{2}$ free parameters that determine the target distribution.

However, as we are only interested in the eigenvalues ϵ_i for the calculation of the LLNs, one might think of strategies for complexity reduction to $O(F)$.

The steps (i) – (vii) cover the complete algorithmic framework of SOFT. However, additional data analysis steps may be performed in order to further investigate the set of trajectory points obtained in step (vii). Among several possible alternatives, one could think of an unsupervised clustering procedure of the vector pairs $(\mathbf{x}(t_d), \mathbf{y}^*(t_f))$ with a low predictability measure $c(t_f)$. The resulting codebook vector positions could be characterized by symbolic labels. They would represent feature detectors for re-recognizable regularities extracted from the empirical time sequence data.

In addition, one might think of iterating the whole procedure of steps (i) – (vii). For this purpose, one could perfom a weighted training of the neural network predictor by adjustment of the learning rate with respect to the predictability measure $c(t_f)$ obtained in the preceding iteration step. For this class of algorithms see e.g. [1]. This could focus the neural network predictor resources in order to increase P-predictability in phase space regions in which transient attractors may be expected, thus enabling a better investigation of T-predictability for their detection. By choosing an appropriate heuristic annealing scheme for the relevance of $c(t_f)$ in the neural network function approximator training procedure, the overall effect could lead to "shrinking islands of good predictability" in the phase space, as the iteration proceeds. These could be labelled by symbols as pointed out above.

Although these additional data analysis procedures may be useful, we want to emphasize that they are not an essential part of our algorithm and may be chosen according to the specific structure of the data set or to the scope of an observer's attention. Instead, we want to stress the key idea of the SOFT approach pointed out in steps (i) – (vii): *the comparative evaluation of source and target distributions induced by "blurring" the input of a neural network function approximator.*

Additional details, implementational issues, complexity considerations, and simulation results will be discussed elsewhere [15] including a critical discussion on differences and interconnections to other fields of research related to time sequence analysis and neural networks (see also [14]).

4 Concluding remarks

The SOFT algorithm presented in this paper couples classical domains of neural network research with respect to the problem of self-organized category formation from empirical time sequence data without presumptive knowledge of re-recognizable regularities: function approximation, PCA, and — as a possible extention — vector quantization. A wide scope of algorithmic alternatives and biological motivations has been discussed in the neural network literature for each of these domains. We do not explicitly specify the implementational details of each of these components. Instead, we focus on the problem, how these components can interact in a constructive manner within the mathematical framework provided by the concept of transient attractors. We hope that SOFT may offer a useful contribution to the field of neural network time sequence analysis.

Acknowledgements

This work has been funded by grants from the Hanns-Seidel-Foundation and the German Federal Ministry of Science and Technology (BMBF).

References

[1] S. Amari, N. Murata, K.R. Müller, M. Finke, and H. Yang. Adaptive on-line learning in changing environments. 1996. To appear in *NIPS'96.*

[2] D.R. Dersch. *Eigenschaften neuronaler Vektorquantisierer und ihre Anwendung in der Sprachverarbeitung.* Verlag Harri Deutsch, Reihe Physik, Bd. 54, Thun, Frankfurt am Main, 1996.

[3] M. Fisz. *Probability theory and mathematical statistics.* Wiley, New York, 1963.

[4] C.L. Giles and C.W. Omlin. Learning, representation, and synthesis of discrete dynamical systems in continuous recurrent neural networks. In *Proceedings of the IEEE workshop on architectures for semiotic modeling and situation analysis in large complex systems.* IEEE press, 1995.

[5] F. Girosi and T. Poggio. Networks and the best approximation property. *Biological Cybernetics,* 63:169–176, 1990.

[6] L. Glass and M.C. Mackey. *From clocks to chaos. The rhythms of life.* Princeton University Press, Princeton, New Jersey, 1988.

[7] H. Jaeger. Identification of behaviors in an agent's phase space. Arbeitspapiere der GMD 951, ftp://ftp.gmd.de/ai-research/publications/-1995/jaeger.95.identify. ps.gz, GMD, St. Augustin, 1995.

[8] H. Jaeger. Dynamische Systeme in der Kognitionswissenschaft. *Kognitionswissenschaft,* 5(4), 1996.

[9] H. Jaeger. From continuous dynamics to symbols. 1997. To appear in *Proceedings of the 1st Joint Conference on Complex Systems in Psychology, 'Dynamics, Synergetics, Autonomous Agents'*, Gstaad, Switzerland.

[10] C.X. Ling and R. Buchal. Learning to control dynamic systems with automated quantization. In P.B. Brazdil, editor, *Machine Learning. Proceedings of the ECML-93, Lecture Notes in Artificial Intelligence*, pages 372–377, Berlin, 1993. Springer-Verlag.

[11] K. Petersen. *Ergodic theory.* Cambridge University Press, 1983.

[12] D.E. Rumelhart and J.L. McClelland. Learning internal representations by error propagation. In *Parallel Distributed Processing*, volume I. M.I.T. Press, Cambridge, MA, 1986.

[13] P. Smolensky. Learning internal representations by error propagation. In *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, volume I, pages 194–281. M.I.T. Press, Cambridge, Mass., 1986.

[14] A. Wismüller and D.R. Dersch. Neural network computation in biomedical research: chances for conceptual cross-fertilization. *Theory in Biosciences*, 116(3), 1997.

[15] A. Wismüller, H. Jaeger, D.R. Dersch, H. Ritter, and G. Palm. Self-organization of feature detectors in time sequences (SOFT) – identification of regularities in an agent's phase space. 1997. In preparation.

[16] Y. Yao and W.J. Freeman. A model of biological pattern recognition with spatially chaotic dynamics. *Neural Networks*, 3(2):153–170, 1990.