ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 413-418

Learning Associative Mappings from Few Examples

Jörg A. Walter

Department of Computer Science · University of Bielefeld
D-33615 Bielefeld · Email: walter@techfak.uni-bielefeld.de

Abstract: The rapid creation of high-dimensional, continuous mappings is a challenge for adaptive and learning methods. This paper discusses the *"Parameterized Self-organizing Maps"* (PSOM) as a suitable method to achieve excellent generalization capabilities from a small set of training data, by making use of available topological information. Furthermore the PSOM provides as an important generalization a flexibly usable, *continuous associate memory* which is, unlike most other existing approaches, not limited to the representation of input-output mappings. This allows to deliver several related mappings – coexisting in a single and coherent framework.

We present two application examples from the field of robotics, a field where the acquisition of large amounts of training data is a severe cost factor. Using only 27 data points, a PSOM can learn the inverse kinematics of a robot finger with a mean positioning accuracy of 1 % of the entire workspace.

Task specifications for redundant manipulators often leave the problem of picking one solution from a subspace of possible alternatives. The PSOM approach offers here a flexible and compact form to select from various constraint and target functions previously associated. Here, in particular, the PSOM learns various ways to resolve the redundancy problem for positioning a 4 DOF manipulator.

1 Introduction

Precise sensorimotor mappings between various motor, joint, sensor, and abstract physical spaces are the basis for many robotics tasks. Their cheap construction is a challenge for adaptive and learning methods. However, the practical application of many neural networks suffer from the need of large amounts of training data, which makes the learning phase a costly operation – sometimes beyond reasonable bounds of cost and effort.

In this contribution we present the "Parameterized Self-Organizing Map" (PSOM) approach, which is particularly useful in situation where a high-dimensional, continuous mapping is desired. If information about the topological order of the training data is provided (or can be inferred) only a very small data set is required. In section 2 the PSOM algorithm is derived from Kohonen's SOM and its *continuous associative completion* mechanism explained. In section 3 we report on a PSOM application for solving the forward and backward kinematics for a robot finger.

If numerous degrees of freedom are available one has to pick one configuration of a continuous space of alternatives. Most solutions of this redundancy problem are based on some pseudo-inverse control (for a review see e.g. [2]). However a more flexible solution should provide an ensemble of suitable action strategies and should offer to respond to different types of constraints. Sec. 4 shows how the PSOM contributes an elegant solution.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 413-418

2 From SOMs to PSOMs

Kohonen [3] formulated the *Self-Organizing Map* (SOM) algorithm as a mathematical model of the self-organization of topographic maps, which are found in brains of higher animals. It is build by a two-dimensional array A of processing units or formal "neurons", where each neuron has a reference vector $\mathbf{w_a}$ attached, which points in the embedding input space X. A presented input $\mathbf{x}$ will select the neuron $\mathbf{a}^*$, which has $\mathbf{w_a}$ closest to the given input, i.e. $\mathbf{a}^* = \text{argmin}_{\forall \mathbf{a} \in \mathbf{A}} \|\mathbf{w_{a'}} - \mathbf{x}\|$. By this competition mechanism the input space is tessellated in *discrete* patches – the so-called *Voronoi cells*. The distribution of reference vectors is obtained by the means of the Kohonen learning rule (see e.g. [3, 6]), which generates a dimension reducing, topographic mapping from a high-dimensional input space to a m-dimensional index space of neurons in the array S.

The main question which led to the PSOM appraoch is, how can the network learn a smooth continuous input–output mapping? The wellknown extensions are the supervised learning of constant output or a local linear map (LLM) [6] within the Voronoi-cells of each neuron $\mathbf{a}$. However, in general the outputs do not match at the cell-borders, which leaves discontinuities in the overall mapping.

The PSOM concept [5] can be seen as the generalization of the SOM with the following three main extensions: *(i)* the discrete index space S in the Kohonen map is generalized to a *continuous mapping manifold* $S \in \mathbb{R}^m$; *(ii)* the embedding space X is formed by the Cartesian product of the input space and output space $X = X^{in} \otimes X^{out} \subset \mathbb{R}^d$; *(iii)* a *continuous mapping* $\mathbf{w}(\cdot) : \mathbf{s} \mapsto \mathbf{w}(\mathbf{s}) \in M \subset X$ is defined, where $\mathbf{s}$ varies continuously over S.

We require that the *embedded manifold* M passes through all supporting reference vectors $\mathbf{w_a}$ and write $\mathbf{w}(\cdot) : S \to M \subset X$ as weighted sum:

$$\mathbf{w}(\mathbf{s}) = \sum_{\mathbf{a} \in \mathbf{A}} H_{\mathbf{a}}(\mathbf{s})\, \mathbf{w_a} \tag{1}$$

This means that, we need a "basis function" $H_{\mathbf{a}}(\mathbf{s})$ for each formal neuron or "node", weighting the contribution of its reference vector (= initial "training point") $\mathbf{w_a}$ depending on the location $\mathbf{s}$ relative to the node position $\mathbf{a}$, and also *all* other nodes $\mathbf{A}$ (however, we drop in our notation the dependency $H_{\mathbf{a}}(\mathbf{s}) = H_{\mathbf{a};\mathbf{A}}(\mathbf{s})$ on $\mathbf{A}$).

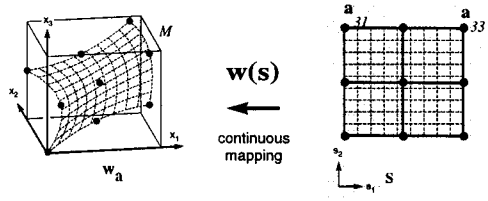

Figure 1: The mapping $\mathbf{w}(\cdot) : S \to M \subset X$ builds a continuous image of the *right* side S in the embedding space X at the *left* side (as indicated by the test grid). The embedding manifold M passes through the reference vectors $\mathbf{w_a}$.

A suitable set of basis functions can be constructed in several ways but must meet two conditions: *(i)* the hyper-surface M shall pass through all desired support points (*orthonormality*), i.e. at those points, only the local node contributes $H_{\mathbf{a}_i}(\mathbf{a}_j) = \delta_{ij}$; $\forall\, \mathbf{a}_i, \mathbf{a}_j \in \mathbf{A}$; *(ii)* the sum of all contribution weights must be one: $\sum_{\mathbf{a} \in \mathbf{A}} H_{\mathbf{a}}(\mathbf{s}) = 1$, $\forall \mathbf{s}$ *(partition-of-unity)*.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 413-418

A simple construction of basis functions $H_{\mathbf{a}}(\mathbf{s})$ becomes possible when the topology of the given points is sufficiently regular. E.g. for a multidimensional rectangular grid, the set of functions $H_{\mathbf{a}}(\mathbf{s})$ can be constructed from products of one-dimensional Lagrange interpolation polynomials. See [7] for details.

Specifying for each training vector $\mathbf{w_a}$ a node location $\mathbf{a} \in \mathbf{A}$ introduces a *topological order* between the training points: training vectors assigned to nodes $\mathbf{a}$ and $\mathbf{a}'$, that are adjacent in the lattice $\mathbf{A}$, are perceived to have this specific neighborhood relation. The effect is important to note: it allows the PSOM to *draw extra curvature information* from the training set. Such information is not available within other techniques, such as the RBF approach and is the essential reason for the generalization capabilities of the PSOM.

When M has been specified, the PSOM is used similar to the SOM: given an input vector $\mathbf{x}$, *(i)* find the best-match position $\mathbf{s}^*$ on the mapping manifold S by minimizing the distance function $dist(\cdot)$

$$\mathbf{s}^* = \underset{\forall \mathbf{s} \in S}{\text{argmin}}\ dist\,(\mathbf{w}(\mathbf{s}), \mathbf{x}) \quad \text{with (e.g.)} \quad dist(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{d} p_k\,(x_k - x'_k)^2. \qquad (2)$$

(ii) The surface point $\mathbf{w}(\mathbf{s}^*)$ serves as the output of the PSOM in response to the input $\mathbf{x}$ and can be viewed as an *associative completion* of the input space component of $\mathbf{x}$. The distance function $dist(\cdot)$ in Eq. 2 is chosen as the Euclidean norm applied only to the input components of $\mathbf{x}$ (belonging to X^{in}). Thus, the function $dist(\cdot)$ actually selects the input subspace X^{in}, since for the determination of $\mathbf{s}^*$ and, as a consequence, of $\mathbf{w}(\mathbf{s}^*)$, only those components of $\mathbf{x}$ matter, that are regarded in the distance metric $dist(\cdot)$. A suitable definition will selects all components k with $p_k > 0$ as belonging to the input subspace; output are components k with $p_k = 0$. The mapping direction can be changed (e.g. reversed) by modifying the coefficients $\{p_k\}$.

The discrete best-match ($\mathbf{a}^*$) search in the standard SOM is here generalized to solve the continuous minimization problem. Starting at the closest node $\mathbf{s}_{start} = \mathbf{a}^*$ the $\mathbf{s}^*$ can be found by a few iterations using the Levenberg-Marquardt algorithm [7].

3 Example: Robot Finger Kinematics

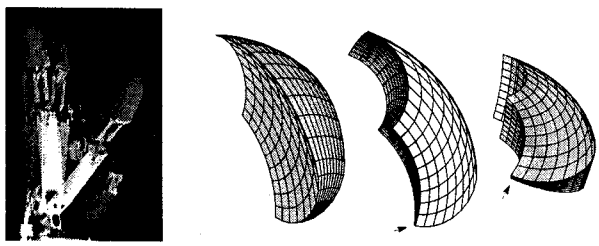

Figure 2: a–d: *(a)* stroboscopic image of one finger in a sequence of extreme joint positions. *(b–d)* Several perspectives of the workspace envelope $\vec{r}$, tracing out a cubical $10\times10\times10$ grid in the joint space $\vec{\theta}$ (the arrow marks the fully adducted position, where one edge contracts to a tiny line).

This section presents the results of applying the PSOM algorithm to the task of learning the kinematics of a 3 degree-of-freedom robot finger of a three-fingered modular hydraulic robot hand, developed by the TU Munich [4]. Its mechanical design allows roughly the mobility

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 413-418

of the human index finger, scaled to by 110%. A cardanic base joint (2 DOF) offers sidewards gyring of $\pm 15°$ and full adduction with two additional coupled joints (1 DOF), see Fig. 2a.

In the case of the finger, there are several coordinate systems of interest, e.g. the joint angles $\vec{\theta}$, the cylinder piston positions $\vec{c}$, one or more finger tip coordinates $\vec{r}$, as well as further configuration dependent values, such as the Jacobian matrices J for force/moment transformations. All these quantities can be simultaneously treated in one single PSOM. Here, we present results of the inverse kinematics, the classical hard part. When moving the three joints on a cubical $10\times10\times10$ grid within their maximal configuration space, the fingertip will trace out the "banana" grid displayed in Fig. 2.

We exercised several PSOMs with n×n×n nine dimensional data tuples $(\vec{\theta}^T, \vec{c}^T, \vec{r}^T)^T$, all equidistantly sampled in $\vec{\theta}$. Fig. 3a–b depicts a $\vec{\theta}$ and an $\vec{r}$ projection of the smallest training set, $n = 3$. To visualize the inverse kinematics ability, we ask the PSOM to back-transform a set of workspace points of known arrangement. In particular, the workspace filling "banana" set of Fig. 2 should yield a rectangular grid of $\vec{\theta}$. Fig. 3c–e displays the actual result. Distortions can be visually detected in the joint angle space *(c)*, and the piston stoke space *(d)*, but disappear after back-transforming the PSOM output to world coordinates *(b)*. The reason is the peculiar structure; e.g. in areas close to the tip a certain angle error corresponds to a smaller Cartesian deviation than in other areas.

When measuring the positioning accuracy we get 1.6 *mm* mean Cartesian deviation, which is **1.0 %** of the maximum workspace length of 160 *mm*. In view of the extremely small training set this appears to be a quite remarkable result.

Nevertheless, this result can be further improved by supplying more training points. For a growing number of network nodes the "Local-PSOM" approach offers to keep the computational effort constant by applying the PSOM algorithm on a sub-grid, see [8, 7].

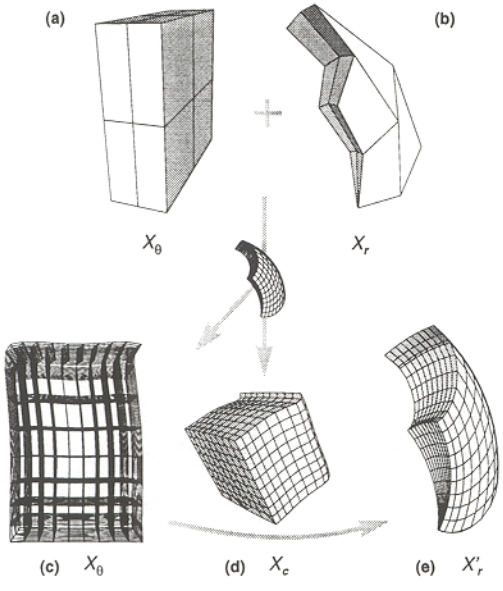

Figure 3: a–b and c–e; Training data set of 27 nine-dimensional points in X for the $3\times3\times3$ PSOM, shown as perspective surface projections of the *(a)* joint angle X_θ and *(b)* the corresponding Cartesian sub space X_r. Following the lines connecting the training samples allows one to verify that the "banana" really possesses a cubical topology. *(c–e)* Inverse kinematic result using the grid test set displayed in Fig. 2. *(c)* projection of the joint angle space X_θ (transparent); *(d)* the stroke position space X_c; *(e)* the Cartesian space $X_{r'}$, after back-transformation.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 413-418

4 Example: Flexible Use of Redundant DOF

In the presence of excess degrees of freedom one has to specify extra constraints to determine a robot configuration. Here, the PSOM's ability to select input sub-spaces offers to extent the embedding space X by various parameterized functions c_j. By adding target values for these, one can utilize the best-match search mechanism to optimize extra goals and constraints.

We illustrate this in the positioning task of a 4 DOF robot in 3D as depicted in Fig. 4 (link lengths 0,1,0.75,0.75 l). For the construction of a PSOM, i.e. the configuration manifold of the arm, we choose an 5×5×5×5 grid covering the joint range $\vec{\theta} \in [0°, 180°]\times[0°, 120°]\times[-120°, 0°]\times[-120°, 0°]$. The embedding space X is spanned by $\mathbf{x} = (\theta_1, \theta_2, \theta_3, \theta_4, r_x, r_y, r_z, c_8, c_9, c_{10})^T$ and contains, similar to the example before, the angles $\theta_{1..4}$, the Cartesian position $\vec{r}$, and, here, three further parameters: c_8 is the difference $(\theta_4 - \theta_3)^2$, c_9 is the elevation angle of link-3 (relative to the horizontal), and c_{10} is the angle between distal link-4 and the vertical.

This allows to resolve the redundancy in various ways. For example, the goal can be to ...
(i) take the minimal joint motion from the current position to the specified position $\vec{r}$: all we need to do is to start the best-match search ($p_{5..7} = 1$) at the best-match position $\mathbf{s}^*_{curr}$ belonging to the current position, and the steepest gradient descent procedure will solve the problem;
(ii) keep joint $j \in \{2, 3, 4\}$ fixed: additionally specify θ_j and $p_j > 0$;
(iii) use similar adduction in the two distal joints (like the finger kinematics): by activating $p_8 > 0$ (to a small value e.g., 0.01) and setting $x_9 = 0$. Measuring the deviation for the inverse kinematics we find a mean value of 0.008 l in the workspace; Fig. 4*(left)* depicts the solution for tracing a circular path for the end effector;
(iv) keep the middle segment horizontal: by specifying the target $x_9 = 0$ and $p_9 = 0.01$. Fig. 4*(middle)* reveals that this constraint can not be met in all cases. By setting p_9 to only a small value, as a "soft goal", the accuracy of the trajectory is not compromised (see also below);
(v) approach vertically: after specifying $x_{10} = 0$, $p_{10} = 0.01$, Fig. 4*(right)* shows the stroboscopic tracking result.

For these different cases we do not need different networks, instead on single PSOM can be utilized. If one anticipates useful target functions, the embedding space can be augmented in advance, enabling to construct reconfigurable optimization modules. They are later activated on demand and show the desired performance. In conflicting situation, e.g. the distal reaching positions in the last example, a meaningful compromise is found. As shown in [7], the input selection coefficients can be made dynamical $p_\mu = p_\mu(t)$ during the iteration process. This enables to formulate priorities of goal functions, e.g. a second rank goal is satisfied as far as

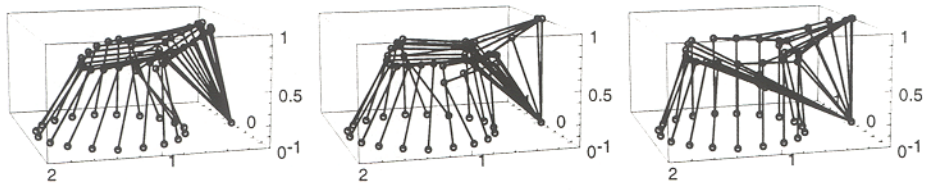

Figure 4: Tracking a point in an horizontal circle with a 4 DOF manipulator, using a 5×5×5×5 PSOM and three different extra target functions to resolve the redundancy problem: *(left)* with maximal similarity of the last two joints; *(middle)* with horizontal middle arm segment; *(right)* with vertical distal arm segment – as far as possible (see distal positions).

the first rank goal (the position goal) is entirely within its solution space.

5 Discussion and Conclusion

We presented the PSOM as versatile building module for learning continuous, high-dimensional mappings. As highlighted by the robot finger example, the PSOM draws its good generalization capabilities from curvature information available through the topological order of only a few reference vectors $\{\mathbf{w_a}\}$. This topological information can be learned by Kohonen's SOM learning rule, or – can be incorporated – if it is known apriori. In many robotic applications, the latter case can be realized by structured sampling of training data – often without extra cost.

The input selection mechanism enables to add further, parameterized target functions. Those can be utilized to resolve e.g. redundancy problems which arise when the primary goal leaves a continuous solution space of possible alternatives. Here the PSOM offers to build a battery of optimizer modules which can be learned within the same continuous associative memory. When they are activated, they can influence the best-match search in the desired manner.

This associative mapping concept has further attractive properties. Several coordinate spaces can be maintained and learned simultaneously (as shown in Fig. 3). This multi-way mapping capability solves, e.g. the forward and inverse kinematics with the very same network. This simplifies learning and avoids worries about inconsistencies of separate learning modules. As pointed out by Kawato [1], the learning of bi-directional mappings is not only useful for the planning phase (action simulation), but also for bi-directional sensor–motor integrated control.

References

[1] Mitsuo Kawato. Bi-directional neural network architecture in brain functions. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN-95), Paris*, volume 1, pages 23–30, 1995.

[2] C.A. Klein and C.-H. Huang. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Trans. Sys. Man and Cybern.*, 13:245–250, 1983.

[3] Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer Series in Information Sciences 8. Springer, Heidelberg, 1984.

[4] R. Menzel, K. Woelfl, and F. Pfeiffer. The developement of a hydraulic hand. In *Proc. of the 2nd Conf. on Mechanotronics and Robotics*, pages 225–238, Sept 1993.

[5] Helge Ritter. Parametrized self-organizing maps. In S. Gielen and B. Kappen, editors, *Proc. Int. Conf. on Artificial Neural Networks (ICANN-93), Amsterdam*, pages 568–575. Springer Verlag, Berlin, 1993.

[6] Helge Ritter, Thomas Martinetz, and Klaus Schulten. *Neural Computation and Self-organizing Maps*. Addison Wesley, 1992.

[7] Jörg Walter. *Rapid Learning in Robotics*. Cuvillier Verlag Göttingen, 1996. also postscript http://www.techfak.uni-bielefeld.de/~walter/pub/.

[8] Jörg Walter and Helge Ritter. Local PSOMs and Chebyshev PSOMs – improving the parametrised self-organizing maps. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN-95), Paris*, volume 1, pages 95–102, 1995.

[9] Jörg Walter and Helge Ritter. Rapid learning with parametrized self-organizing maps. *Neurocomputing*, 12:131–153, 1996.