Neurobiology Suggests the Design of Modular Architectures for Neural Control
Jean-Luc Buessler, Jean-Philippe Urban

To cite this version:

HAL Id: hal-00950496
https://hal.archives-ouvertes.fr/hal-00950496
Submitted on 21 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
J.L. BUESSLER, J.P. URBAN
Neurobiology Suggests the Design of Modular Architectures for Neural Control

Preprint

MIPS-TROP-TR-02-03
Neurobiology Suggests the Design of Modular Architectures for Neural Control

J.L. Buessler       J.P. Urban
TROP Research Group at the University of Mulhouse
4 rue des Frères Lumière, 68093 Mulhouse, France
JP.Urban@univ-mulhouse.fr — www.trop.univ-mulhouse.fr

Abstract

The existence of modular structures in the biological world strongly suggests that the training of this kind of structures is actually feasible. It is a key indication for the development of neural networks applications, especially in the field of robotics. Indeed, a single network can only efficiently treat problems with few independent variables; the combining of several networks is necessary to address more complex tasks.

We investigate learning techniques and show that using a particular form of architecture can ease the training of a modular structure: a bi-directional structure that allows combining several neural networks. The approach is illustrated with Kohonen’s self-organizing maps for a robotic visual servoing task.

1. Introduction

Neurobiologists collect more and more precise data on the working of the CNS (Central Nervous System). The brain appears as a highly modular structure. Recent reviews on visually guided reaching [1,2] reveal a distributed functioning over multiple areas. There is no unique sensorial space perception preceding action, as has been supposed for along time. A multiplicity of areas brings about distinct sensorimotor transformations, each area using specific sensorial data [1]. These transformations are realized progressively by strongly functionally related areas linked by reciprocal connections [2].

This information is important for the design of artificial systems. For example, Kawato and his group proposed interesting applications of bi-directional architecture where Artificial Neural Networks (ANN) are linked by reciprocal connections [3].

Our group investigates the possibilities of learning in such modular structures. Adaptation in neural systems appears a priori very difficult. Indeed, the design of architectures combining several ANN stumbles currently on the difficulty of training the various networks. Besides, since the brain presents both a large modularity and an extraordinary learning capability, it is important to search the mechanisms in action.

This article shows that a bi-directional architecture can considerably ease learning and could be a key to modular neural structures. The approach is introduced in a very general manner; no specific assumptions are made on the learning mechanism. The second part of the article describes an implementation with Kohonen’s Self-Organizing Maps, and a robotic arm-positioning task using data supplied by a stereoscopic vision head.

2. Modular Architecture and Learning

2.1. Architectural units

Considering the brain as an aggregation of specialized modules, many kinds of combinations can be considered. Two basic architectural units are used to illustrate the problems of learning.

Figure 1: An example of parallel organization: mixture-of-experts

---

Jacobs and Jordan’s mixture-of-experts [4] represents an example of parallel organization (see Figure 1). Each neural network is considered as an expert. It can carry out the task by itself, but is mainly efficient on a fraction of the input space. A gating network weighs out the experts’ responses to determine the global output response.

On the other hand, a modular decomposition can break up the treatment into a sequence (see Figure 2). One module carries out part of the treatment. Its response is an intermediate result transmitted to another module. The interest of this architecture becomes considerable when sub-tasks can be defined that will only treat a fraction of the variables. This modularity reduces thus the dimensionality of the modules to be implemented. Figure 3 describes the basic architectural motif. Input vector \( x \) is partitioned into two sets \( x_1 \) and \( x_2 \) (not necessarily disjoint).

Multiple variants of units and composition are possible. Each individual module can itself be decomposed.

### 2.2. Supervised learning

To examine the compatibility of these modular structures with a learning capability, the nature of this learning must be defined. Our approach is developed with a stochastic supervised learning scheme, but the reasoning can be generalized to other training variants.

Supervised learning is defined in terms of example-based adaptation of system parameters (i.e. synaptic weight). It is stochastic when the data are used online, without being stored. The weights are iteratively updated with randomly presented examples that depend on the interactions of the system with its environment.

To define the notations more formally, a neural network responds with \( \hat{y} = R(x) \) to input vector \( x \). The neural function can also be denoted \( \hat{y}_k = R(x_k,W_k) \) to show the dependence of the response to parameters \( W_k \). Presenting example \( (x_k,y_k) \), the weights are modified to reduce error \( \epsilon_k = y_k - \hat{y}_k = y_k - R(x_k,W_k) \), or a cost function \( J(\epsilon_k) \).

In the parallel organization of Figure 1, each expert can use examples \( (x_k,y_k) \). They compete to learn the training patterns. The gating network mediates the competition and is itself adapted. Jacobs and Jordan developed these learning algorithms [4].

In a sequential structure, on the other hand, the examples are not sufficient anymore to supervise the adaptation of each of the modules. The intermediate variables \( z_k \) introduce new degrees of freedom in the system that are not defined in the response model (see figure 3).

Today, there is no practical solution available for the learning of such a modular structure. And thus, this type of modular structure cannot be used in ANN applications, whose use is limited to problems of small dimensionality. The composition can only be implemented if each network can be trained independently. Ingenious application examples can be found in robotics (e.g. [5]).

### 2.3. Internal representations

The systematic interconnection reciprocity that is noticed within the brain areas suggests an extension of the neural module concept. Instead of a single ANN, we consider that each module contains a second network \( R_{inv} \) associated to the first, that learns the inverse correlation, from the outputs to the inputs. This extension defines a data flow of opposed direction.

To simplify the analysis, we assume that function \( y = F(x) \), that is to be learned by the neural structure, can be decomposed in the form:

\[
y = F_2(z,x_2) \quad \text{with} \quad z = F_1(x_1)
\]

and there exists a function \( G(z) \) such as:

\[
z = G(y)
\]

Functions \( F_1, F_2, \) and \( G \), are smooth. They are not necessarily known. They only establish the possibility to carry out learning. For example, supervised learning of module \( M_i \) has only meaning if \( z \) is defined and unique for a given vector \( x_i \).

The neural structure thus reproduces a mathematically possible decomposition and can provide a precise response if the training of both modules is carried out suitably.
It is easy to see on Figure 4 that module 2 is entirely supervised with data known at instant $k$: $(x_{2,k}, \hat{z}_{2k}, y_k)$. We can define the adaptation of modules $R_1$ and $R_{2inv}$ by taking each other’s response as desired output to minimize the difference between their estimations.

Consider the class of solutions of functional equation:

$$R_1(x_{1,k}) = R_{2inv}(y_k)$$

where $R_1(x_{1,k}) = R_{2inv}(z_{2k}, x_{2k})$ using equation (1)

Since the relation must stand for all $x_1, x_2$, the second term depends only on $z$ and thus represents a function $h(z)$. Function $h$ is in this case a scalar function of $z$. If $h$ can be inverted, then $\hat{z} = h(z)$ can be defined as an internal representation of variable $z$. The training phase leads the neural networks to represent respectively:

$$R_1(x_{1}) = h(F_1(x_{1}))$$
$$R_2(\hat{z}, x_{2}) = F\gamma(h^{-1}(\hat{z}), x_{2})$$
$$R_{2inv}(y_k) = h(G(y_k))$$

In this configuration, the third network serves only to ease learning phase. The composition of the two first realize the desired general function:

$$\hat{y} = R_2(\hat{z}, x_{2}) = R_2(R_1(x_{1}), x_{2}) = F_{\gamma}(F_1(x_{1}), x_{2}) = F(x)$$

### 2.4. Regularization

Any invertible function $h$ will fit the relation above. It is therefore not necessary to impose the choice of the internal representation, for example $h(z) = z$. However, one must verify that the adaptation does not converge to a trivial solution $h(z) = constant$. One can also take advantage of the freedom of representation to favor a smooth function with an amplitude range compatible with the elements of $x_2$. The adaptation of the networks will be more efficient.

The regularization techniques are difficult to implement in the context of stochastic learning. The most efficient approach seems here to be an algorithm introduced by J.F. Cardoso and B. Laheld [6], to adapt iteratively a whitening matrix.

The training examples are adapted for one of the networks, for example $R_1$, which will no more take $(x_{1,k}, \hat{z}_{2,k})$ as model, but $(x_{1,k}, \hat{z}_{2,k})$ with

$$\hat{z}_k = v_k \cdot (\hat{z}_{2,k} - m_k)$$

where $v_k$ and $m_k$ correspond respectively to a whitening factor and a mean estimation of $\hat{z}_{2,k}$. These two variables allow to enforce a constraint of zero mean and unit variance for the learning flow of network $R_1$. They are iteratively estimated with $\hat{z}_{2,k}$, using equations:

$$m_{k+1} = m_k + \gamma m (\hat{z}_{2,k} - m_k)$$
$$v_{k+1} = v_k + \gamma v_k \left(1 - (v_k \hat{z}_{2,k})^2\right)$$

where $\gamma_m$ and $\gamma_v$ are learning rates comprised between 0 and 1. They can be time varying.

### 2.5. Bi-directional architecture

This learning approach, based on bi-directional data flows, can be easily generalized. It uses exclusively local data, that belong to the considered neural stage. The adaptation technique is identical when the considered modules belong to a structure that gathers multiples stages. Its principle is simple and allows to design more complex variants and modular architectures.

The design of these architectures is based on the possible decompositions of data processing and the necessity to reduce the dimensionality of the subtasks. In many applications, a judicious data organization permits to meet the constraints expressed in equations (1) and (2).

The example in the following section shows that learning is really feasible and efficient when several bi-directional stages are chained and produce several internal representations.

### 3. An example of composition

The application we introduce uses several Self-Organizing Maps (SOM) to represent the motor and sensorial position correlations of a robotic platform. Two active cameras follow the movements of a robot manipulator in 3-D space. The mapping of image positions and camera orientations into arm angular joint

---

**Figure 4:** Bi-directional extension of the learning diagram depicted in Figure 3. The regularization variables appear symbolically on Module 1.
The active vision head can be defined using three maximum workspace (see Figure 5). The experimental setup is composed of a robotic platform. The cameras of a robotic vision system's performances. The cameras of a robotic head are continuously oriented in direction of the interest zone.

3.1. Visual and robotic task

We consider a robotic arm in movement in its 3-D workspace and study the control with visual information. Active vision techniques are implemented to ensure the vision system's performances. The cameras of a robotic head are continuously oriented in direction of the interest zone.

3.2. Supervised Self-Organizing Maps

Among neural networks algorithms, we choose Kohonen’s Self-Organizing Maps, and its extension by Ritter et al. [7]. The algorithm is simple, well-suited for online adaptation, and converges rapidly.

The SOM algorithm is based on 3 major elements:
- a pre-defined grid of fixed dimension where each unit (neuron) \( r \) is defined by its position \( \mathbf{r} \),
- the distance between neurons \( r \) and \( s \) and the neighborhood function \( h_{rs} = h(r,s) \),
- weight vectors, associated to each unit \( r \): \( \mathbf{w}^m_r \) in the input space, \( \mathbf{w}^o_{out} \) in the output space.
At a given time instant, the competition between the units within the layer results in a winner \( s \), the neuron whose weights best fit the input vector \( x \):

\[
|x - w_{in}^s| \leq |x - w_{in}^r|, \quad \forall r
\]

This neuron activates the associated output vector:

\[
\hat{y} = w_{out}^s
\]

The learning step for the input and output weight vectors are respectively:

\[
w_{in}^r = w_{in}^r + \varepsilon h(s)(x - w_{in}^r)
\]

\[
w_{out}^r = w_{out}^r + \varepsilon' h'(s)(y - w_{out}^r)
\]

\( \varepsilon \) and \( \varepsilon' \) are learning rates and \( h(s) \) and \( h'(s) \) are neighborhood functions of the form:

\[
h(s) = \exp \left( \frac{|s - s_f|}{2\sigma^2} \right)
\]

The values of the parameters \( \varepsilon \), \( \varepsilon' \), \( \sigma \) and \( \sigma' \) decrease with the number of learning steps.

### 3.3. Neuro-controller architectures

In a previous work [8], we analyzed the decomposition into two modules whose structure is depicted in Figure 5. This architecture is briefly described, and then a complementary decomposition of Module B is proposed. The results presented here confirm the working of a structure that chains 4 bi-directional modules.

**a) Bi-directional Learning**

The diagram in Figure 5 depicts the two steps involved in the arm motor command based on active vision data: \( G: [\alpha, V] \rightarrow \alpha^F \) and \( H: \alpha^F \rightarrow \theta \). This modular decomposition is interesting because it is based on an internal representation which corresponds to \( \alpha^F \), the robot head commands to fixate the 3-D target point with the cameras.

The elements of vector \( \alpha^F \) can be expressed as the sum \( \alpha^F = \alpha + \Delta \alpha = \alpha + f(V) \). We consider that the camera orientation variation is only a function of vector \( V \), and not of the current orientation. This is an acceptable approximation that is necessary for the dimensional reduction of the data to be treated.

Module A uses a 3-D SOM map. The adaptation of this module is based on the bi-directional learning principle. Module B combines two 3-D maps, one for the direct data flow, the other for the inverse flow.

The learning parameters are detailed in [8].

**b) Extended modularity** (figure 6)

Module A use an additive composition of 2 maps, respectively of dimensions 1-D and 2-D.

An analysis of the robot’s geometrical invariances leads to the decomposition of Module B into a 3-stage bi-directional structure. The neuro-controller’s global architecture is then based on 8 SOM maps, seven 2-D and one 1-D. With this thorough decomposition, we will verify that the advantage of having dimensionally reduced maps is not annihilated by the induced more complex modular structure.

**c) Comparison tests**

Two tests are proposed to give elements of comparison. The **independent learning** assumes that the data necessary to perform supervised independent adaptation of the two modules are available. This test provides an ideal learning reference. The second reference, the **single network** test, attempts to learn \( F: [V, \alpha] \rightarrow \theta \) with a single SOM map. It shows the failure of this approach when too many independent variables are involved.

---

**Figure 6: Extended modularity.** Module A uses an additive composition of two maps. Module B is broken up into three 2-D bi-directional modules. Each module Bi reproduces the structure of Module B in Fig. 5.
3.4. Results and discussion

The main purpose of the presented results is to validate the learning technique. The modular architecture is compatible with rapid learning and good convergence. We present the results of experiments conducted on a real robotic platform. Table 1 presents the positioning error for a test set after a learning phase (10,000 presented examples). The mean error value is close to the minimum, for the considered discretization, when the training of the networks can be supervised independently (independent learning test). In the general case, in which the data are not available to supervise each network, the bi-directional learning technique shows its efficiency, even when the decomposition introduces several modular stages.

Table 1: Architecture comparison and performance

<table>
<thead>
<tr>
<th>Evaluation after 10,000 training cycles</th>
<th>Positioning error mean (max) in cm</th>
<th>Number of SOM networks</th>
<th>Number of SOM neurons</th>
<th>Response computing time (with learning) ms/sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single neural network</td>
<td>10.1 (31)</td>
<td>1 (3-D)</td>
<td>420,000</td>
<td>860</td>
</tr>
<tr>
<td>Independent learning</td>
<td>2.6 (10)</td>
<td>2 (3-D)</td>
<td>84,000</td>
<td>190</td>
</tr>
<tr>
<td>Bi-directional structure (fig 5)</td>
<td>3.0 (11)</td>
<td>3 (3-D)</td>
<td>126,000</td>
<td>280</td>
</tr>
<tr>
<td>Extended modularity (fig 6)</td>
<td>2.9 (10)</td>
<td>8 (2-D)</td>
<td>15,800</td>
<td>45</td>
</tr>
</tbody>
</table>

Figure 7 presents the error evolution during the training of the extended modular structure. The resulting curve is very similar to the one obtained with the supervised adaptation of each of the SOM maps.

The interest of the modular approach appears clearly with the examination of the performances displayed by the single network test. Although a large number of neurons are used, it does not converge on a satisfying representation of the function. Besides, when modularity permits to reduce the dimensionality of the networks, and therefore the number of neurons implied for an equivalent discretization, the computational cost is greatly reduced.

4. Conclusion

The bi-directional learning technique enables the creation of an internal representation that does not have to be supervised during the training. It provides the means to design modular architectures and to break up the treatments.

The experiments have been presented using supervised SOM networks. They converge rapidly, but result in a relatively coarse space discretization. The presented bi-directional technique is however not dependent on a given learning algorithm.

5. References