Modeling Cortical Maps with Feed-Backs

Thierry Viéville and Pierre Kornprobst

***Abstract*— High-level specification of how the brain represents and categorizes the causes of its sensory input allows to link "what is to be done" (perceptual task) with "how to do it" (neural network calculation). More precisely, a general class of cortical map computations can be specified representing what is to be done as an optimization problem, in order to derive the related neural network parameters considering regularization mechanisms (implemented using so-called partial-differential-equations).**

The present contribution revisits this framework with three add-ons. It is generalized to a larger class of (non-linear) map computations, including winner-take-all mechanisms. The capability to represent standard "analog" neural network and guaranty their convergence, providing their weights are local and unbiased, is made explicit. The fact that not only *one* but several cortical maps can interact, with feed-backs, in a stable way is shown.

Two experiments are provided as an illustration of this general framework.

I. Introduction

Perceptual processes architecture, in computer or biological vision [1], [2], is based the computation of "maps" of quantitative values. Such maps encode retinotopic quantities such as contrast magnitude, contrast orientation related to edge orientation, shape curvature, binocular disparity related to the visual depth, color cues, temporal disparity between two consecutive images in relation with visual motion detection, etc. Other maps are not only parametrized by retinotopic locations, but also other parameters (e.g. orientation, retinal velocity, etc.) or more abstract quantities [3].

Using a scalar or vector valued map is an important feature when addressing the modelization of cortical processing units such as cortical columns [2]. It may also help defining improved models of neurons or small neuronal assemblies, where the state is not only defined by a scalar membrane potential [4].

Introducing non-linear constraints between the map components has several advantages, one is to take noisy measures into account avoiding statistical bias (see e.g. [5] for a development), another is to define physical parameters (e.g. 3D orientation) with complex structure.

High-level specification [1], [6], [4] of such map computation, as reviewed for instance in [7], consider the estimation process as a mechanism which iteratively corrects the output in order to correctly predicts the input (i.e. with *Expectation* which "infers" the output from the given inputs and *estiMation* which "predicts" the input from "a-priory" estimates of output). In the Bayesian framework (but not only) this correspond to a criterion optimization, e.g. finding the "maximally probable output given the input.

Thierry.Vieville@sophia.inria.fr, INRIA BP93, 06902 Sophia, France

In the present approach, the map computation problem is formalized as a minimization problem define in equation (2). The reader is advised to consider the previous references for further details.

We start in Section II by introducing main definitions. In Section III, we propose to model one cortical map by variational formulation in the continuous setting. In Section IV, we show how to pass from this continuous variational formulation to the related discrete neural network. In Section V we consider the reciprocal result, i.e. how to relate a given neural network to a variational apporach? Section VI extends this framework it is shown that not only one map but a graph of maps can be specified in this framework to several cortical maps, under some biologicaly plausible assumptions. Sections VII and VIII revisit some existing approaches under the proposed framework, with some illustrations.

II. Map Input, Ouput, Sampling and Connectivity

Let us introduce main definitions and assumptions. The specifications and the derivations of the proposed framework are explictly based on these choices so we have to make them explicit first.

The goal of neural map computation is to obtain an ouput map $\mathbf{v} : \mathbb{R}^n \rightarrow \mathbb{R}^p$, from an input map $\mathbf{w} : \mathbb{R}^n \rightarrow \mathbb{R}^q$, based on a variational approach. For example, for $n = 2$, the space R^2 could be a representation of the retina domain, $\mathbf{w}$ the retina intensity with $q = 1$ (or $q = 3$ if color) and $\mathbf{v}$ the motion estimation. We assume that the functions $\mathbf{v}$ and $\mathbf{w}$ belong to a dense linear subset of an Hilbert space H, more precisely the Sobolev space $H = W^{s,\infty}(\mathbb{R}^n)$. Note that vectors, and also matrices, are written in bold characters, matrices with capital letters and scalars in italic.

In Section III we consider a continous framework because this is the simplest way to specify the map computation. This is however not exactly the gound truth of the neural units.

Sampling: In fact, if considering the micro-columns of a cortical map [2], there is a clear sampling of the underlying continuous quantities. In Section IV we deal with such neural network implementing map computations, only defined at a finite set of positions $\mathbf{y}_j \in \mathbb{R}^n$. The value of the map $\mathbf{v}$ at $\mathbf{y}_j$ is a measure in a small neighborhood $\mathcal{S}_j$ around $\mathbf{y}_j$ defined by

$$\mathbf{v}_j = \mathbf{v}(\mathbf{y}_j) = \int_{\mathcal{S}_j} \mathbf{v}(\mathbf{y})\, \mu_j(\mathbf{y})\, d\mathbf{y},$$

where $\mu_j(\mathbf{y})$ is the measure density in $\mathcal{S}_j$. Up to a simple scale factor and without loss of generality, we assume that $\int_{\mathcal{S}_j} d\mathbf{y} = 1$.

Two classical choices are $\mu_j(\mathbf{y}) = \delta(\mathbf{y} - \mathbf{y}_j)$ [8] or $\mu_j(\mathbf{y}) = 1$ [9] which gives an average measure. Results presented hereafter do not depend on the choice of μ_j.

Connections: For one neuronal position $\mathbf{x} \in \mathbb{R}^n$, we assume it is connected to a finite set of M samples $\{\mathbf{y}_1, \ldots, \mathbf{y}_j, \ldots, \mathbf{y}_M\}$ in a neighborhood $\mathcal{S}$ around $\mathbf{x}$.

Considering a biological neural network, it is important, due to the huge complexity of the underlying mechanisms, to consider the weakest assumption about how each sample neighborhoods. More precisely consider overlapped neighborhoods $\mathcal{S}_j \cap \mathcal{S}_i \neq \emptyset$ or partial partitioning $\cup_j \mathcal{S}_j \nsubseteq \mathcal{S}$ are allowed (see Fig. 1).

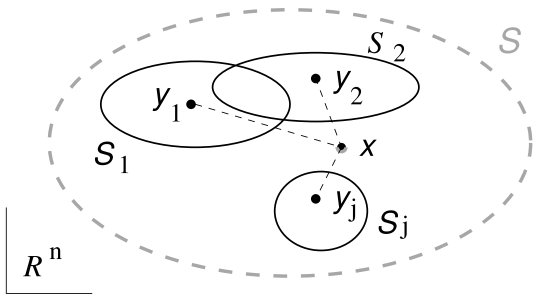


Fig. 1. Schematic representation of the sample neighborhoods.

In order to related the continuous specification (defined in Section II) to the discrete implementation (detailed in Section III) based on such connectivity we must relate the discrete measures $\mathbf{v}(\mathbf{y}_j)$ to the underlying continuous quantity. In the derivation of the main result in [7] reported here, this link relies on the following *summation property* defining a measure $\mu(\mathbf{y})$:

$$\begin{aligned} \mathbf{v}(\mathbf{x}) = & \int_{\mathcal{S}} \mathbf{v}(\mathbf{y})\, \mu(\mathbf{y})\, d\mathbf{y} \\ = \textstyle\sum_j & \int_{\mathcal{S}_j} \mathbf{v}(\mathbf{y})\, \mu_j(\mathbf{y})\, d\mathbf{y} + \int_{\mathcal{S} - \cup_j \mathcal{S}_j} \mathbf{v}(\mathbf{y})\, \mu_\bullet(\mathbf{y})\, d\mathbf{y}, \end{aligned} \tag{1}$$

defining $\mu_\bullet(\mathbf{y})$ as the measure density where no sample is available. This formula simply states that measures are linear related, i.e. that the different samples are combined additively. It is verified by any sampling model [8], [10], [11], [9], although often implicitly and not at this level of generality.

III. Specification of Cortical Map Computation

According to generative approaches [1], [6], [4], the cortical map computation can be modeled as an optimization problem. Let us state this in the very general form proposed in [7].

Given an input map $\mathbf{w}$, one look for an output map $\bar{\mathbf{v}}$ verifying

$$\bar{\mathbf{v}} = \underset{\mathbf{v} \in H / \mathbf{c}(\mathbf{v}) = 0}{argmin} \; \mathcal{L}(\mathbf{v}), \quad \text{with} \tag{2}$$

$$\mathcal{L}(\mathbf{v}) = \int |\hat{\mathbf{w}} - \mathbf{w}|^2_{\mathbf{\Lambda}} + \int \phi(|\nabla \mathbf{v}|_{\mathbf{L}}) + \int \psi(\mathbf{v}), \tag{3}$$

$$\text{and } \hat{\mathbf{w}} = \mathbf{P}\, \mathbf{v}, \tag{4}$$

where ∇ stands for the gradient operator, $\phi(\cdot)$, $\psi(\cdot)$, $\mathbf{P}$, $\mathbf{c}(\cdot)$, $\mathbf{\Lambda}$ and $\mathbf{L}$ are commented hereafter. The norms defined in (3) are weighted norms defined by $|\mathbf{u}|_{\mathbf{M}} = \mathbf{u}^T \mathbf{M} \mathbf{u}$, where $\mathbf{M}$ is, when $\mathbf{u}$ is a vector, a symmetric positive matrix, while where $\mathbf{M}$ is a tensor if $\mathbf{u}$ is a matrix.

Figure 2 is a representation of the model (2)–(4). The first term in (3) is a fidelity attached term specifying how the output is related to the input, the second term is a smoothing term which defines the regularity of the output and the third term allows to constraint the form of the solution. The equation (4) shows the choosen relation between the estimation of the input given an output. So the formulation (2)–(4) specifies the cortical map computation in the sense that it explains the "goal", what is to be done, but without any reference to how it is done. The rest of this section is devoted to the analysis of each term.

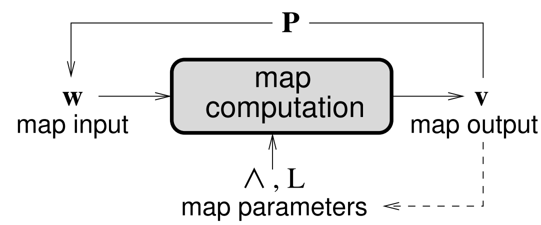


Fig. 2. Cortical map computation: how to obtain $\mathbf{v}$ from $\mathbf{w}$? Note that the output is used to modify model parameters and to have an estimation of the input map.

The functions $\mathbf{\Lambda} : \mathbb{R}^n \rightarrow S_m^+ H$, where S_m^+ is the set of square symmetric positive semi-definite matrices of size m, defines a so-called *measurement information metric* which represents

- The *precision of the input*: the higher this precision in a given direction, the higher the value of $\mathbf{\Lambda}$ in this direction (in a statistical framework, $\mathbf{\Lambda}$ corresponds to the inverse of a covariance matrix);
- *Partial observations and missing data*: if the input is only defined in some directions, it corresponds to a matrix $\mathbf{\Lambda}$ definite only in these directions (e.g. if only defined in the direction $\mathbf{u}$, $\mathbf{\Lambda} = k\, \mathbf{u}\, \mathbf{u}^T$ for some k), if the input is missing we simply have to state $\mathbf{\Lambda} = 0$.

The functions $\mathbf{L} : \mathbb{R}^n \rightarrow T^{n \times n}_{p \times p}$, where $T^{n \times n}_{p \times p}$ is the set of twice symmetric matrix metric tensors, defines a *diffusion tensor* $\mathbf{L}$, which is symmetric and positive (i.e. $\forall \mathbf{M}, \mathbf{M}^T\, \mathbf{L}\, \mathbf{M} \geq 0$). The weighted norm of $\nabla \mathbf{v}$ is modulated by a function $\phi : \mathbb{R} \rightarrow \mathbb{R}$ which controls the amount of smoothness required. For example, $\phi(s) = s^2$ is called a Tikhonov penalty term: it strongly penalizes variations of $\mathbf{v}$ (high cost in the energy for high gradients) so that the resulting $\mathbf{v}$ will be oversmoothed. If one want to preserve edges, i.e. the discontinuities of $\mathbf{v}$, it is necessary to choose a smoothing term less penalizing. Several ϕ functions have been proposed (see [12] for a review and discussion): the function $\phi(s) = \sqrt{1 + s^2}$, convex and with linear growth at infinity. Not only the smoothing is weaker but also it allows to have smooth solutions in homogeneous regions, with sharp discontinuities. Furthermore, $\mathbf{L}$ may be a function of $\mathbf{v}$ defining a feed-back as developed in the sequel.

When a problem is ill-posed, i.e. if there are many (and usually numerically unstable) solutions, adding some a priori on the smoothness of the solution is the key idea to have a problem well-posed. When the input function is partially or approximately defined at some points, as discussed previously, the value at such a point is defined using information "around" which diffuses (as discussed now) from well-defined values to undefined or ill-defined values.

Three kinds of constraints are introduced in (2)–(4):

- *Structural constraints*, written $\mathbf{c}(\mathbf{v}) = 0$, force the solution to belong belong to a manifold defined by implicit equations. For example, to represent an orientation $\theta \in [-\pi..\pi]$ we consider $\mathbf{v} = (p, q)$ with $p = \cos(\theta)$ and $q = \sin(\theta)$ well-defined by the constraint $\mathbf{c}(\mathbf{v}) = p^2 + q^2 - 1 = 0$. This Euclidean embedding of an orientation allows one to estimate p and q without considering parametrization issues around $\pm\pi$. So the proposed framework is very general regarding non-linear object representations (see e.g. [5] for a general discussion).
- *Optimization constraints* (via the $\psi(v)$ term of the criterion) simply allows to weakly constraint v to get closer to a given set of solutions (e.g. the binarization term in the winner-take-all mechanism experimented in the sequel).
- *Measurement constraints* between both the input and the quantity to estimate (via the "measurement" relation $\hat{\mathbf{w}} = \mathbf{P}\,\mathbf{v}$). It is known that in order to obtained an unbiased estimation (e.g. [5]) the measure itself has to be estimated, which corresponds to integrate $\hat{\mathbf{w}}$ in the estimation, thus in $\mathbf{v}$, as made explicit via the relation linear relation $\mathbf{P}$.

In the cortex, such "neuronal unit" is a cortical hypercolumn. Our model can be be mapped onto usual computational model of cortical columns processes (see also [2] for a treatise on the subject). Regarding such a "processing a cortical map being a discrete implementation of such neural units network (see e.g. [13]), we propose in table I a possible interpretation of such an abstract analog network. This mapping is to be understood as a working assumption. It also make explicit the scale at which such analog networks should be situated. This mapping has the chance to be

$\mathbf{w}$	Extra cortical input or intra-cortical input from previous layers
$\mathbf{v}$	Extra cortical or backward intra-cortical output
σ_{ij}	Local connections weights
$\mathbf{\Lambda}, \mathbf{L}$	Remote backward connections
Iterative operations	Internal connections

Variables of the models can be interpreted with respect to a cortical column connectivity.

TABLE I

compatible with the laminar architecture of the cortex or neocortex [14], [15] and with the related inter-layer circuitry. Excitatory but also inhibitory connectivity is included in the diffusion term as detailed in [9].

IV. The Network Compilation Rules

The solution of (2)–(4) can be implemented using a discrete network as defined in the following proposition.

Proposition 4.1: The optimization problem (2)–(4) *is, in the general case, locally minimized by the following linearized differential equation:*

$$\frac{\partial \mathbf{v}_i}{\partial t} = -\epsilon_i(\mathbf{v}_i) + \sum_j \sigma_{ij}(\mathbf{v}_i)\,\mathbf{v}_j + \kappa_i\,\mathbf{w}_i \tag{5}$$

with:

$$\begin{cases} \epsilon_i(v) &=& \rho_i\,\mathbf{v} + \xi\,\frac{\partial \mathbf{c}}{\partial \mathbf{v}}^T \mathbf{c} + \frac{\partial \psi}{\partial \mathbf{v}}^T, \\ \rho_i &=& \sum_j \sigma_{ij} + \mathbf{P}^T\,\mathbf{\Lambda}_i\,\mathbf{P}, \\ \kappa_i &=& \mathbf{P}^T\,\mathbf{\Lambda}_i, \end{cases} \tag{6}$$

and $\xi = (1-\lambda)\,|\frac{\partial \mathcal{L}}{\partial \mathbf{v}}|/|\frac{\partial \mathbf{c}}{\partial \mathbf{v}}^T \mathbf{c}|$ *with* $\lambda \ll 1$. *The weights* $\sigma = (\sigma_{ij})$ *are given by considering the linearized optimal integral approximation up to order* r ($r \geq 2$) *of the non-linear diffusion operator*

$$\bar{\mathbf{L}} = \phi'(|\nabla \mathbf{v}|_{\mathbf{L}})\,\mathbf{L}, \tag{7}$$

defined at M *points, providing* $M > \frac{(n+r)!}{n!\,r!} - \frac{n\,(n+1)}{2}$. *They are given by solving the systems:*

$$\begin{array}{rcl} \bar{\mathbf{L}}_{kl}(\mathbf{x}_i) &=& \frac{1}{2}\sum_j \sigma_{ij}\,\bar{\mu}_j^{\mathbf{e}_k+\mathbf{e}_l}(\mathbf{x}), \\ \mathbf{div}_k(\bar{\mathbf{L}}(\mathbf{x}_i)) &=& \sum_j \sigma_{ij}\,\bar{\mu}_j^{\mathbf{e}_k}(\mathbf{x}), \end{array} \tag{8}$$

where[1]*:* $\bar{\mu}_j^{\alpha}(\mathbf{x}) = \int_{\mathcal{S}_j} (\mathbf{y}-\mathbf{x})^{\alpha}\,\mu_j(\mathbf{y})\,d\mathbf{y}$, *while:*

$$\forall i, \quad \sum_j \sigma_{ij}\,\bar{\mu}_j^{\alpha}(\mathbf{x}) = 0 \quad 2 < |\alpha| \leq r \tag{9}$$

Among all σ_{ij} *verifying* (8) *and* (9) *we choose those which verify (here* $|\sigma_{ij}|^2$ *represents the* $\mathcal{L}^2$ *norm):*

$$\mathit{min} \sum_{ij} |\sigma_{ij}|^2, \tag{10}$$

■

What is meant by general in the previous proposition is the fact (5) relates on the Euler-Lagrange conditions of (2) in which the integral approximation of the diffusion operator has been introduced. Time t is thus related to the convergence of the minimization. This is a necessary condition which leads to the solution of (2) at the convergence (see e.g. [12] for details on such methods).

Integral approximation of the diffusion operator: Integral approximation of differential operators have been introduced in the field of neural networks by Cottet et al. [16], [17], [10], [11] and presented in the present form in [7], where the derivation of the present proposition is available. In particular, the present form provides an alternative to the use of so-called particles methods (e.g. [8]) which neural interpretation is weaker.

It is important to note, as demonstrated in [8], that this integral approximation is not only closed to the related differential operator, but that it also leads to sampled solutions

[1]We use the standart multi-index notation, for vector of integer indices $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathcal{N}^n$ and $|\alpha| = \alpha_1 + \ldots + \alpha_n$ and $\mathbf{x}^{\alpha} = x_1^{\alpha_1} \ldots x_n^{\alpha_n}$

which are closed to the continuous solutions. This is due to the fact $\mathbf{L}$ is a positive operator.

Regarding the fact we limit the approximation up to order r, since for a constant K:

$$\bar{\mu}_j^\alpha(\mathbf{x}) \leq K\,[|\sigma_{kl}^\epsilon|_{0,\infty}/(|\alpha|+1)]\,\epsilon^{|\alpha|+1},$$

as shown in [11], $\bar{\mu}_j^\alpha$ becomes arbitray small so that *unbiasness* constraints (9) are automatically verified up to a negligible quantity when $|\alpha|$ increases.

In the form of the previous proposition, these approximations are based on the summation property (1) and provide a direct link between a discrete integral approximation and the continunous differential operator, without the introduction of a "discretization" step. This is an improvement with respect to [11], [8].

In fact, not only the optimal approximation defined by (10), but a *whole family of kernel* (the linear sub-space defined by (8) and (9) implements a diffusion operator, including unbounded kernels [8], thus allowing to represent a rather large class of networks as detailled in the next section.

Implementation of the network: Since the present integral approximation is obtained from a quadratic minimization (10) with linear constraints (8) and (9), it is well-defined and the solution is a closed-form linear function of $\bar{\mathbf{L}}$ and $\mathbf{div}(\bar{\mathbf{L}})$. Furthermore, this linear function is only defined by the network sampling, because only function of $\bar{\mu}_j^\alpha(\mathbf{x}_i)$. Furthermore, it appears that the coefficients σ_{ij} can also easily derived from a Hebbian rule (see [7] for details).

As a consequence, the network parameters defined in (6) are directly given in closed-form from the specification equations $\mathbf{P}$, $\mathbf{c}(\cdot)$, $\psi(\cdot)$ and parameters $\mathbf{\Lambda}$ and $\mathbf{L}$. They are "compilable" and:

- ϵ_i contains a positive leakage term and corrective terms related to the constraint to verify;
- κ_i acts as a gain product proportional to the input reliability as in usual, less general analog networks.

As made explicit in [7], the coefficient ξ must be *small* enough to decrease $\mathcal{L}$ and *high* enough to maintain $\mathbf{c}(\mathbf{v}) = 0$, which is obvious to adjust numerically, avoiding the explicit computation of the related formula. When non-linear constraints are not considered, $\xi = 0$.

V. Representation of Analog Network

Let us now detail to which extends we can relate a given analog network to a criterion of the form (2)–(4).

Proposition 5.1: Given a network dynamic of the form

$$\frac{\partial \mathbf{u}_i}{\partial t} = -\epsilon_i(\mathbf{u}_i) + \sum_j \sigma_{ij}(\mathbf{u}_i)\,\mathbf{v}_j + \kappa_i\,\mathbf{w}_i \qquad (11)$$

with $\mathbf{v}_i = Sig(\mathbf{u}_i)$, as soon as the weigths σ_{ij} are unbiased, i.e. verify (9), locally minimizes in the general case the criterion:

$$\int |\hat{\mathbf{w}} - \mathbf{w}|^2 + \int |\nabla \mathbf{v}|^2_{\mathbf{L}} + \int \psi(\mathbf{v}), \qquad (12)$$

with $\hat{\mathbf{w}} = \kappa^T\,\mathbf{v}$, $\psi = \int \epsilon - [\sum_j \sigma_j + \mathbf{P}^T\mathbf{P}]\,\mathbf{v}$ and $\mathbf{L}$ defined by (8). ■

This result is applicable to analog Hopfield network (as detailled e.g. in [11]) and to the very powerful class of models Cohen and Grossberg dynamical system (e.g. [18], [19]) for which it has been shown [7] that the previous result is applicable.

Here a sigmoidal non-linearity is introduced between the neuronal state input and ouput as illustrated in Fig. 3. The result is in fact true with or without this non-linear term [7].

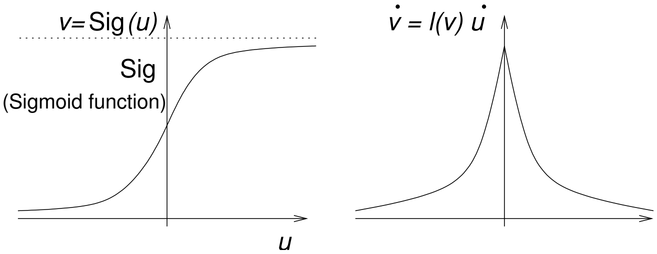


Fig. 3. In analog networks a sigmoidal non-linearity (denoted by Sig) is introduced between the neuronal state $\mathbf{u} \in \mathbb{R}^N$ (usually related to the membrane potential) and the neuronal output $\mathbf{v} \in [0,1]^N$ (usually related to the average firing rate probability).

One add-on of this specification is that convergence of the network is demonstrated without the restrictive assumption of symmetry of the weights σ_{ij} (e.g. [18]).

This representaton is well-defined for neural network connectivity with short-range connections, since this connectivity implements a local diffusion operator. Network mechanisms based on remote connections are not expected to have unbiased weigths σ, thus are not correctly represented in this framework.

VI. Stable Interaction Between Maps

The cortex can be considered as a hierarchy of cortical levels with reciprocal extrinsic cortico-cortical connections among the constituent cortical areas. This notion of a hierarchy depends upon a distinction between forward and backward extrinsic connections. This distinction rests upon different laminar specificites (see e.g. [20], [6] for a review). Main properties are summarized for the visual system in table II.

It is important to emphasize that forward/backward connection is not only an "anatomical" but also a "functional" distinction. Furthermore, feedback from one area to another use backward connections, eventually in the very early part of the processing of a given input [20]. As a consequence, given an input, backward connections are not necessarily acting "after" forward connections but as soon as a forward connection feed a backward one.

Let us formalize these interactions, considering several cortical maps $\bar{\mathbf{v}}_m$ and defined as previously:

$$\bar{\mathbf{v}}_m = \underset{\mathbf{v}_m \in H/\mathbf{c}_m(\mathbf{v}_m)=0}{argmin} \mathcal{L}_m(\mathbf{v}_m),$$

Forward connections	Backward connections
Sparse axonal bifurcation	Frequent bifurcation
Patchy axonal terminations	Diffuse axonal terminations
Topographic projections	Non-topographic projections
One-to-one / small divergence	Large spatial divergence
Define a lattice	Transcend several levels
	Slow time-constants
	More numerous

Forward connections are ”driving” for promulgation and segregation of sensory information. Bacward connections are ”modulatory” for mediation of contextual effects, co=ordination of processing. This table describe their main caracteristics.

TABLE II

from (2)–(4) all quantites being indexed by the map index m.

Here, each cortical map value can tune the other map computations, modifying the parameters $\mathbf{\Lambda}_m$ and $\mathbf{L}_m$. The key point is that we use what has been observed in the cortex (e.g. [6]) to introduce three constrains:

- Feedback values are smoothed in space, before influencing other maps,
- A cortical map value modifies its state and may also its own parameters,
- Forward connections define a lattice (thus without loop).

Let us write $\mathcal{S}$ the smooth operator. Here we consider a spatial smoothing. In practice $\mathcal{S}$ is likely a spatio-temporal smoother, with a small delay temporal low-pass filter. From the biology, the assumption is that feed-back are able to induce very rapid changes thus induce only very small delays. The main result proposed here is compatible with the introduction of such a delay.

Furthermore, a rectification function $\rho(\cdot)$ is also introduced, e.g. $\rho(u) = max(u, 0)$ to take into account the fact that only positive quantities is output by cortical maps.

The vector $\mathbf{v}_\bullet = (\cdots \mathbf{v}_m, \cdots)$ is the concatenation of all cortical maps values. It also corresponds to the *backward* connections onto any map, since we do not introduce any restriction at this stage.

We thus formalize feed-back connections considering $\mathbf{\Lambda}_m(\mathcal{S} * \rho(\mathbf{v}_\bullet))$ and $\mathbf{L}_m(\mathcal{S} * \rho(\mathbf{v}_\bullet))$ as continuous derivable functions.

A feed-forward connection from a cortical map of index m to a cortical map of index m' corresponds to the fact that $\mathbf{w}_{m'} = \rho(\mathbf{v}_m)$. Our understanding of the cortical maps interactions is that they form a graph with no cycle.

The situation is very different when backward connections interact: several criteria are to be minimized simultaneously, yielding a apparently very complex dynamical system, with the risk of interferences, oscillations, chaotic behavior, etc..

Thanks to biologically plausible properties introduces here, a solution of this problem is based on the following fact.

Proposition 6.1: Locally minimizing the criteria $\mathcal{L}_m$ with respect to $\mathbf{v}_m$ is equivalent to locally minimize with respect to $\mathbf{v}_\bullet$, in the general case:

$$\mathcal{L}_\bullet = \sum_m \lambda(|\nabla_m \mathcal{L}_m|)\, \mathcal{L}_m \tag{13}$$

writing $\nabla_m = \partial/\partial \mathbf{v}_m$ and $\lambda(\cdot) : \mathcal{R}^+ \rightarrow \mathcal{R}^+$ a positive strictly increasing profile with

$$\lambda(u) \geq 0, \lambda'(u) > 0, \lambda(0) = 0 \;\; \textit{and} \;\; \lim_{u \to 0} \lambda'(u)/u < +\infty.$$

(e.g. $\lambda(u) = u^\alpha$ with $\alpha > 2$).

Furthermore, locally minimizing each criterion $\mathcal{L}_m$ using (5) locally minimizes the common criterion $\mathcal{L}_\bullet$ as soon as the functions $\psi_m(\cdot)$ are convex. ∎

The criterion defined in (13) provides a view of what is a common objective for the different cortical maps computations.

This is a crucial fact, because it means that minimizing each criterion is a convergent process, since it corresponds to a common criterion minimization. As a consequence, we have a formal verification that feedback links in our framework yields a well-defined process.

The fact $\psi(\cdot)$ is convex restrains the specification to specific criteria and appears to be an important requirement as visible in the derivation of the proposition.

This derivation is given in the appendix and is based on two remarks:

(1) In the *absence* of backward connections, the final result is easy to predict: given some inputs, each iterative computation in a cortical map yield a stable result and from upstream to downstream this stable result propagates. This is the case for very fast brain computation [21] where, due to very short latencies, only feed-forward computations occur.

(2) When backward connections are active, the convergence is preserved by the fact that spatial smoothing allows to neglect possible perturbations from feedback loops onto the minimization process.

The previous derivation also describes qualitatively how interactions between different cortical maps occurs:

- Backward connections have a constant influence in the sense that they can very rapidly tune the processing of a cortical map but do not interfere with the convergence inside a such a map, they propagate information between the cortical maps, in a stable way; very fast propagation can occur in “one step”, i.e. without inducing transient effects;
- Forward connections act as a “data propagation” though the related lattice and may induce transient effects on downstream layer;
- If a cortical map input is changed (because the cortex inputs vary dynamically) the overall process is still convergent.

VII. Example 1: Edge-Preserving Smoothing

Let us revisit an edge-preserving smoothing approach proposed by Cottet and Ayyadi [10] which corresponds to the framework presented in this paper. In [10], given an initial image $w : \mathbb{R}^2 \rightarrow \mathbb{R}$, the authors proposed a diffusion processes of the form :

$$\frac{\partial \mathbf{v}}{\partial t} = l(\mathbf{v}) \, \Delta_{\mathbf{L}(\mathbf{v})} \mathbf{v}$$

where $l = 1/Sig^{-1}$, which is in fact related to the minimisation of the criterion

$$\bar{v} = \underset{v}{argmin} \; \mathcal{L}(v) = \lambda \int (w - v)^2 + \int |\nabla v|^2_{\mathbf{L}}, \qquad (14)$$

where λ is a small constant and $\mathbf{L}$ is defined by

$$\mathbf{L} = \left[\rho^2 \, \mathbf{P}_{\mathbf{g}^\perp} + \frac{3}{2} \, (1 - \rho^2) \, \mathbf{I} \right] \qquad (15)$$

$$\text{with} \quad \begin{cases} \mathbf{g} = S \star \nabla v, \\ \rho = \min \left(1, \frac{|\mathbf{g}|^2}{s^2} \right), \\ \mathbf{P}_{g^\perp} = \begin{pmatrix} g_2^2 & -g_1 g_2 \\ -g_1 g_2 & g_1^2 \end{pmatrix}, \end{cases}$$

where s is the contrast threshold, τ is an adaptation time constant and S is a spatial smoothing kernel. $\mathbf{P}_{\mathbf{g}^\perp}$ is the 2D projection onto $\mathbf{g}^\perp$, thus on the edge tangent, $\mathbf{g}$ being aligned with the edge normal direction. Depending on the norm of the gradient of the intensity, the smoothing term will infer two kinds of behaviors:

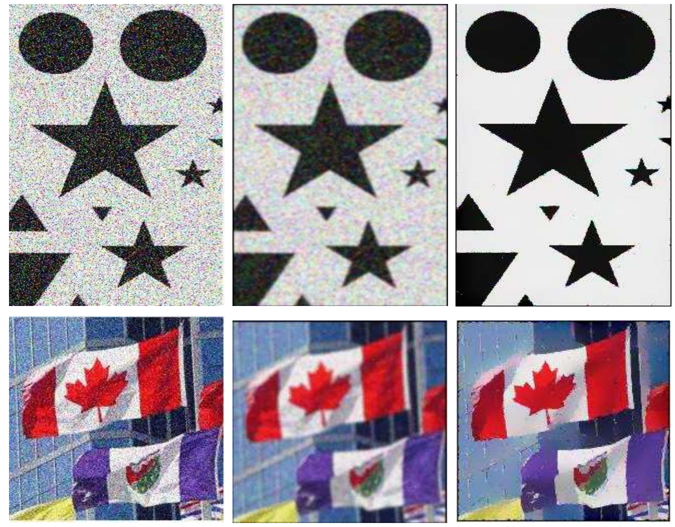
- For low contrasts, when ρ is close to zero, we have $\mathbf{L} \equiv \mathbf{I}$: the smoothing term is quadratic which corresponds to an isotropic smoothing in the Euler-Lagrange equation.
- For high contrasts, when ρ is close to one, we have $\mathbf{L} \equiv \mathbf{P}_{\mathbf{g}^\perp}$: the smoothing term will perform anisotropic diffusion only in the normal direction to the edges.

Fig. 4 shows some comparison of this adaptive linear diffusion process compared with classical linear diffusion. Thanks to the short-term adaptation of the diffusion tensor $\mathbf{L}$ discontinuities are preserved. The adaptive rule (15) corresponds to a Hebbian rule at the implementation level [10], and it can interpreted as a feedback link from previous estimation of v onto the forward diffusion process (see the dotted arrow in Fig. 2).

It has been formally shown [10] that combining short-term adaptation with the diffusion process is a convergent process: the key point is that the feedback from v to $\mathbf{L}$ is smooth in space and time.

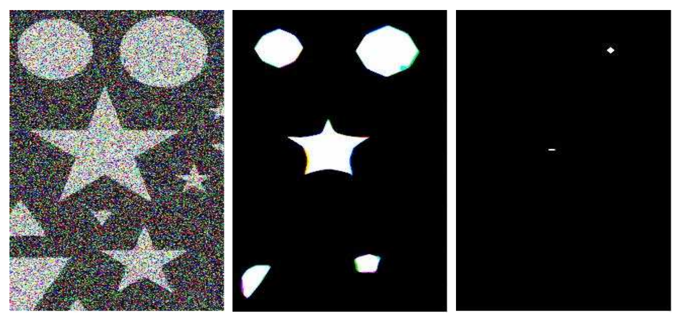
Note that contrarily to [10],we have not introduced here a non-linearity as discussed for (11): we have implemented a linear neural-network as in (5). We verified experimentaly that this non-linearity is not determinant and does not influence significantly the resulting image.

At step ahead, in [10], a temporal filtering is introduced in the feedback. Thus, it is not directly $\mathbf{L}$ but an exponential temporal filtering of $\mathbf{L}$ which is taken into account. Proposition 6.1 prediction is that such a low-pass filtering is not required and we have been able to verify this fact in this context. More precisely, we have experimented that a small-delay (0 .. 10 times the sampling period) low-pass filter does not significantly influence the result, whereas higher delays inhibit the feedback, inducing a convergence with only a poor edge-preserving smoothing.



Initial image Isotropic Anisotropic

Fig. 4. Two examples of results using anisotropic diffusion (right image), the 1st example being the same as in [10] to validate the present method. The original image is on the left. As a comparison, a Gaussian filtering (isotropic diffusion) is shown in the middle. The synthetic image contains a huge (80%) amount of noise. The real image contains features at several scales. In both cases edges are preserved, while an important smoothing has been introduced (from [9]).



Input *Intermediate* *Output*

Fig. 5. Two examples of result for the winner-take-all mechanism implemented using the proposed method. The very noisy (more than 80%) original image is on the left; the intermediate result shows how diffusion is combined with erosion yielding the final result, shown also with a zoom. Clearly the focus is given on the main structures of the image, We have experimented a correct behavior on many different inputs.

VIII. Example 2: The Winner-Take-All (WTA)

Let us now describe how WTA mechanism can be written in this framework.

WTA mechanisms are usually realized (e.g. [22], [23], [18]) using an ad-hoc mechanism with an explicit definition of inter-neuron inhibition in order to allow one neuron to maintain its activity whereas all other activites vanish. They are used in may neuronal computations (see the review in [22]) and the way they could be implemented is still an issue. It is thus a important test for the present method to verify if such a mechanism is easily formalized.

Given an initial condition w, one look for a solution $\bar{v}$

verifying

$$\bar{v} = \underset{v}{argmin}\ \mathcal{L}(v) = \int (w-v)^2 + \int |\nabla v|^2 + \int \psi(v) \quad (16)$$

where $\psi : [0,1] \rightarrow \mathbb{R}$ is a bi-modal function, for example

$$\psi(v) = v^{2\,t/(1-t)}(1-v)^2,$$

with $\psi(0) = \psi(1) = 0$ and $\phi'(t) = 0$ in fact maximal at $t \in]0,1[> 1/2$. This previous expression is the simplest polynomial profile with the suitable craracteristics: this non-linear term will force the values of the network to be zero or one, with a bias towards the zero value.

Formulation (16) is directly related to the general framework (2–4) where no constraint is used and $P \equiv I$.

In Fig. 5 an example of result is shown, with an adaptive profile $\psi'(\cdot)$ (the threshold t is initialized to the distribution mean and incremented/decremented during the process to maintain a small binarization with respect to diffusion). The iteration is stopped when the output has a predefined small size.

This very simple mechanism shows how the present formalism may provide a complementary view with respect to other analog network approaches [23], [18].

IX. Conclusion

In this paper we revisited the links between high-level specification of how the brain represents and categorizes the causes of its sensory input and related analog. We represented what is to be done as an optimization problem with regularization terms and showed how to to compile the related analog or spiking neural network parameters. Although requiring non-negligible derivations, the present technical developments are rather elementary in the sense that we deliberately avoid using the theory of functional spaces to specify and derive our results.

It appears that not only analog networks but also deterministic spiking neural network can be linked to the present specifications. With piece-wise approximations in the so called Spike Response Model [24], it has been possible to map equation (5) of the present framework on a spiking neural network

- The resistive coefficient ρ_i being proportional to the spiking threshold;
- The weights σ_{ij} being in direct relation with the synaptic weights;
- The corrective term ϵ_i being controlled by the axonal delay;
- The input gain being κ_i controlled by the input resistance;

with closed-form correspondence allowing to explicitly calculate the neural network parameters given an abstract continuous representation. This also leads to a fast event-based simulation of such networks. This is a perspective of this work, drafted in [7] following the work of [25], [26] and it will be the next step regarding the present development.

Acknowledgment: This work is realized within the scope of the European FACETS project in direct relation with scientific contributions of partners of this consortium.

References

[1] T. Lee and D. Mumford, “Hierarchical bayesian inference in the visual cortex,” *J. Opt. Soc. Am. A*, vol. 20, no. 7, 2003.
[2] Y. Burnod, *An adaptive neural network: the cerebral cortex*. Masson, Paris, 1993, 2nd edition.
[3] T. Gisiger, S. Dehaene, and J. P. Changeux, “Computational models of association cortex,” *Curr. Opin. Neurobiol.*, vol. 10, pp. 250–259, 2000.
[4] P. Dayan and L. F. Abbott, *Theoretical Neuroscience : Computational and Mathematical Modeling of Neural Systems*. MIT Press, 2001.
[5] T. Viéville, D. Lingrand, and F. Gaspard, “Implementing a multi-model estimation method,” *The International Journal of Computer Vision*, vol. 44, no. 1, 2001. [Online]. Available:
[6] K. Friston, “Functional integration and inference in the brain,” *Prog Neurobiol*, vol. 68, pp. 113–143, 2002.
[7] T. Viéville, “An abstract view of biological neural networks,” INRIA, Tech. Rep. RR-5657, Aug. 2005. [Online]. Available:
[8] P. Degond and S. Mas-Gallic, “The weighted particle method for convection-diffusion equations,” *Mathematics of Computation*, vol. 53, no. 188, pp. 485–525, 1989.
[9] T. Viéville, “An unbiased implementation of regularization mechanisms,” *Image and Vision Computing*, vol. 23, no. 11, pp. 981–998, 2005. [Online]. Available:
[10] G.-H. Cottet and M. E. Ayyadi, “A Volterra type model for image processing,” *IEEE Transactions on Image Processing*, vol. 7, no. 3, Mar. 1998.
[11] R. Edwards, “Approximation of neural network dynamics by reaction-difusion equations,” *Mathematical Methods in the Applied Sciences*, vol. 19, pp. 651–677, 1996.
[12] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, ser. Applied Mathematical Sciences. Springer-Verlag, Jan. 2002, vol. 147. [Online]. Available:
[13] J. Petitot and Y. Tondut, “Vers une neuro-géométrie. fibrations corticales, structures de contact et contours subjectifs modaux,” *Mathématiques, Informatique et Sciences Humaines*, vol. 145, pp. 5–101, 1999.
[14] D. Mumford, “On the computational architecture of the neocortex. i. the role of the thalamo-cortical loop,” *Biological Cybernetics*, vol. 65, pp. 135–145, 1991.
[15] R. Douglas and K. A. C. Martin, “Neuronal circuit of the neocortex,” *Ann. Rev. Neuroscience*, vol. 27, p. 419, 2004.
[16] G. Cottet and S. Mas-Gallic:, “A particle method to solve the navier-stokes system,” *Numer. Math.*, vol. 57, 1990.
[17] G.-H. Cottet, “Neural networks: continuous approach and applications to image processing,” *J. Biological Systems*, vol. 3, 1995.
[18] S. Grossberg, “Nonlinear neural networks: Principles, mechanisms, and architectures,” *Neural Networks*, vol. 1, no. 1, pp. 1–97, 1988.
[19] R. Raizada and S. Grossberg, “Towards a theory of the laminar architecture of the cerebral cortex: Computational clues from the visual system,” *Cerebral Cortex*, vol. 13, pp. 100–113, 2003.
[20] J. Bullier, “Integrated model of visual processing,” *Brain Res. Reviews*, vol. 36, pp. 96–107, 2001.
[21] S. Thorpe and M. Fabre-Thorpe, “Seeking categories in the brain,” *Science*, vol. 291, pp. 260–263, 2001.
[22] A. J. Yu, M. Giese, and T. Poggio, “Biophysiologically plausible implementations of maximum operation,” *Neural Computation*, vol. 14, no. 12, 2003. [Online]. Available:
[23] H. Frezza-Buet, N. Rougier, and F. Alexandre, “Integration of biologically inspired temporal mechanisms into a cortical framework for sequence processing,” in *Neural, Symbolic and Reinforcement methods for sequence learning*, ser. Lecture Notes in Computer Science. Springer, 2001. [Online]. Available:
[24] W. Gerstner and W. Kistler, *Spiking Neuron Models*. Cambridge University Press, 2002.
[25] W. Maass, “Fast sigmoidal networks via spiking neurons.” *Neural Computation*, vol. 9, pp. 279–304, 1997.
[26] W. Maass and T. Natschler, “Networks of spiking neurons can emulate arbitrary hopfield nets in temporal coding,” *Neural Systems*, vol. 8, no. 4, pp. 355–372, 1997.

Appendix

Let us derive the proposition 6.1. In (i) we show that if all $\mathcal{L}_m$ are minimal so is $\mathcal{L}_\bullet$, in (ii) we derive the reciprocal statement, in (iii) we show that, when no forward links, applying (5) minimizes $\mathcal{L}_\bullet$ and in (iv) we finally show that it is still true with forward links.

(i) if all $\mathcal{L}_m$ are minimal with respect to $\mathbf{v}_m$, then $\nabla_m \mathcal{L}_m = 0$ thus $\lambda(|\nabla_m \mathcal{L}_m|) = 0$ and $\mathcal{L}_\bullet = 0$, but $\mathcal{L}_\bullet$ is also positive as a sum of positive terms, so that 0 is its minimal value, now attained: $\mathcal{L}_\bullet$ is thus minimal.

(ii) If $\mathcal{L}_\bullet$ is minimal, its gradient vanishes, i.e. $\nabla_\bullet \mathcal{L}_\bullet = (\cdots \nabla_m \mathcal{L}_\bullet, \cdots) = 0$, writing $\nabla_\bullet = \partial / \partial \mathbf{v}_\bullet$ but:

$$\begin{array}{rcl} \nabla_m \mathcal{L}_\bullet & = & \sum_n \lambda(|\nabla_n \mathcal{L}_n|) \, \nabla_m \mathcal{L}_n + \\ & & \lambda'(|\nabla_n \mathcal{L}_n|) \nabla_m (|\nabla_n \mathcal{L}_n|) \, \mathcal{L}_n \end{array}$$

Let us rewrite this condition :

$$\forall m \quad 0 = \nabla_m \mathcal{L}_\bullet = \lambda(|\nabla_m \mathcal{L}_m|) \, \nabla_m \mathcal{L}_m + \gamma_m$$

with $\gamma_m = \sum_{n \neq m} \lambda(|\nabla_n \mathcal{L}_n|) \, \nabla_m \mathcal{L}_n + \sum_n \lambda'(|\nabla_n \mathcal{L}_n|) \nabla_m (|\nabla_n \mathcal{L}_n|) \, \mathcal{L}_n$.

These two vectors $\nabla_m \mathcal{L}_m$ and γ_m have a "huge" dimension

$$dim(\nabla_m \mathcal{L}_m) = dim(\gamma_m) = dim(\mathbf{v})$$

i.e. the dimension of the map (the number of neurons). In the general case, there are independent, because γ_m contains, if generic, terms which do not linearly depends on $\nabla_m \mathcal{L}_m$. As a consequence, their sum being equal to zero, both vectors must vanish so that

$$\lambda(|\nabla_m \mathcal{L}_m|) \, \nabla_m \mathcal{L}_m = 0 \Rightarrow \nabla_m \mathcal{L}_m = 0$$

The gradients of all $\mathcal{L}_m$ vanish, so that these quadratic criteria are locally minimized.

(iii) Let us write:

$$\begin{array}{rcl} \mathcal{L}_m & = & \int |\mathbf{P} \, \mathbf{v}_m - \mathbf{w}_m|^2_{\mathbf{\Lambda}_m(\mathcal{S} * \rho(\mathbf{v}_\bullet))} + \\ & & \phi(|\nabla \mathbf{v}_m|^2_{\mathbf{L}_m(\mathcal{S} * \rho(\mathbf{v}_\bullet))}) + \psi(\mathbf{v}_m) \end{array}$$

From a few algebra:

$$\nabla_m \mathcal{L}_n = \nabla^\epsilon_m \mathcal{L}_n + \left\{ \begin{array}{l} \mathbf{P}^T \, \mathbf{\Lambda}_m \, [\mathbf{P} \, \mathbf{v}_m - \mathbf{w}_m] - \Delta_{\bar{\mathbf{L}}_m} \mathbf{v}_m + \nabla \psi(\mathbf{v}_m) \\ \qquad \text{if } m = n \\ \rho'(\mathbf{v}_m) \, \mathbf{\Lambda}_n \, [\mathbf{w}_n - \mathbf{p}(\mathbf{v}_n)] \\ \qquad \text{if } \mathbf{w}_n = \rho(\mathbf{v}_m) \\ 0 \\ \qquad \text{otherwise} \end{array} \right.$$

writing $\nabla^\epsilon_m \mathcal{L}_n = \frac{1}{2} \left[|\mathbf{P} \, \mathbf{v}_n - \mathbf{w}_n|^2_{\nabla_m \mathbf{\Lambda}_n} + |\nabla \mathbf{v}_m|^2_{\nabla_m \mathbf{L}_n} \right]$ while $\bar{\mathbf{L}}$ has been defined in (7).

Because of the action of the spatial smoothing operator:

$$\begin{array}{rcl} \nabla_m \mathbf{\Lambda}_n & = & \nabla[\mathbf{\Lambda}_n(\mathcal{S} * \mathbf{v}_{\bullet n})] \\ & = & \nabla \mathbf{\Lambda_n}(\mathcal{S} * \mathbf{v}_{\bullet n}) \, [\nabla \mathcal{S} * \mathbf{v}_{\bullet n}] \end{array}$$

has a small magnitude since, by definition, of a smoothing operator $|\nabla \mathcal{S}|$ is small, say $|\nabla \mathcal{S}| = o(1/\omega)$, where ω is the smoothing filter window size (e.g. for an isotropic Gaussian filter $\mathcal{S}(\mathbf{u}) = 1/\sqrt{2\,\pi}/\omega e^{-\frac{1}{2}\frac{|\mathbf{u}|^2}{\omega^2}}$ the reader can easily verify that $|\nabla \mathcal{S}| = o(1/\omega^{1 + dim(\mathbf{v})/2})$. This is also true if $\mathcal{S}$ contains some small temporal filtering, since the gradient $|\nabla \mathcal{S}|$ is still negligible in this case, from the previous arguments.

Similarly $\nabla_m \mathbf{L}_n = o(1/\omega)$. As a linear combination of small quantities, $\nabla^\epsilon_m \mathcal{L}_n$ is thus small and we can write:

$$|\nabla^\epsilon_m \mathcal{L}_n| = o(1/\omega)$$

A step further, from (5) and (6):

$$\begin{array}{l} \nabla^2_m \mathcal{L}_m \\ = \nabla_m \left[\mathbf{P}^T \, \mathbf{\Lambda}_m \, [\mathbf{P} \, \mathbf{v}_m - \mathbf{w}_m] \right. \\ \left. \quad - \Delta_{\bar{\mathbf{L}}_m} \mathbf{v}_m + \nabla \psi(\mathbf{v}_m) \right] + o(1/\omega) \\ \simeq \nabla_m \left[\mathbf{P}^T \, \mathbf{\Lambda}_m \, [\mathbf{P} \, \mathbf{v}_m - \mathbf{w}_m] \right. \\ \left. \quad - \sum_j \sigma_{n.j} \mathbf{v}_{nj} + \sum_j \sigma_j \, \mathbf{v}_m + \nabla \psi(\mathbf{v}_m) \right] + o(1/\omega) \\ = \mathbf{P}^T \, \mathbf{\Lambda}_m \, \mathbf{P} + \sum_j \sigma_j + \nabla^2 \psi(\mathbf{v}_m) + o(1/\omega) \end{array}$$

Here $\nabla^2_m \mathcal{L}_m$ is the sum of positive symmetric matrices, since $\nabla^2 \psi$ is positive assuming $\psi(\cdot)$ is convex, while $\sum_j \sigma_j$ is positive because $\bar{\mathbf{L}}$ is positive [7].

Yet another step further:

$$\begin{array}{l} \nabla_m \mathcal{L}_\bullet \\ = \sum_n \lambda(|\nabla_n \mathcal{L}_n|) \, \nabla_m \mathcal{L}_n + \lambda'(|\nabla_n \mathcal{L}_n|) \nabla_m (|\nabla_n \mathcal{L}_n|) \, \mathcal{L}_n \\ = \sum_n \lambda(|\nabla_n \mathcal{L}_n|) \, \nabla_m \mathcal{L}_n + \frac{\lambda'(|\nabla_n \mathcal{L}_n|)}{|\nabla_n \mathcal{L}_n|} \, \nabla_n \mathcal{L}_n \, \nabla_m \nabla_n \mathcal{L}_n \\ = \sum_n \lambda(|\nabla_n \mathcal{L}_n|) \, \nabla_m \mathcal{L}_n + \frac{\lambda'(|\nabla_n \mathcal{L}_n|)}{|\nabla_n \mathcal{L}_n|} \, \nabla_n \mathcal{L}_n \, \nabla_n \nabla_m \mathcal{L}_n \end{array}$$

this expression being well defined even for small $\nabla_n \mathcal{L}_n$, i.e even when closed to the optimum, because $\lim_{u \rightarrow 0} \lambda'(u)/u < +\infty$.

Without forward link (i.e. no (m, n) with $\mathbf{w}_n = \rho(\mathbf{v}_m)$) we have noticed that when $n \neq m$, $|\nabla_m \mathcal{L}_n| = |\nabla^\epsilon_m \mathcal{L}_n| = o(1/\omega)$ so that the previous expression reduces in this case to:

$$\begin{array}{rcl} \nabla_m \mathcal{L}_\bullet & = & \lambda(|\nabla_m \mathcal{L}_m|) \, \nabla_m \mathcal{L}_m + \\ & & \frac{\lambda'(|\nabla_m \mathcal{L}_m|)}{|\nabla_m \mathcal{L}_m|} \nabla_m \mathcal{L}_m \, \nabla^2_m \mathcal{L}_m + o(1/\omega) \end{array}$$

Let us rewrite (5) as:

$$\frac{\partial \mathbf{v}_m}{\partial t} = -\nabla_m \mathcal{L}_m^T - \beta_m$$

where $\beta_m \equiv \frac{\partial \mathbf{c}_m}{\partial \mathbf{v}_m}^T \mathbf{c}_m(\mathbf{v}_m)$ has been calculated when deriving (5) to maintain:

$$\nabla_m \mathcal{L}_m \, (\nabla_m \mathcal{L}_m^T + \beta_m) > 0$$

while $\beta_m \rightarrow 0$, because (5) drives $\mathbf{c}_m(\mathbf{v}_m) \rightarrow 0$ whatever the minimization process is [7].

This yields:

$$\begin{array}{rcl} -\frac{\partial \mathcal{L}_\bullet}{\partial v} & = & -\nabla_\bullet \mathcal{L}_\bullet \, \frac{\partial \mathbf{v}_\bullet}{\partial t} \\ & = & -\sum_m \nabla_m \mathcal{L}_\bullet \, \frac{\partial \mathbf{v}_m}{\partial t} \\ & = & \sum_m \nabla_m \mathcal{L}_\bullet \, (\nabla_m \mathcal{L}_m^T + \beta_m) \\ & = & \sum_m \left[\lambda(|\nabla_m \mathcal{L}_m|) \, \nabla_m \mathcal{L}_m \, (\nabla_m \mathcal{L}_m^T + \beta_m) \right. \\ & + & \left. \frac{\lambda'(|\nabla_m \mathcal{L}_m|)}{|\nabla_m \mathcal{L}_m|} \nabla_m \mathcal{L}_m \, \nabla^2_m \mathcal{L}_m \, (\nabla_m \mathcal{L}_m^T + \beta_m) \right] \\ & & + o(1/\omega) \end{array}$$

As a consequence, without forward links, the previous expression is a sum of positive terms, so that $-\frac{\partial \mathcal{L}_\bullet}{\partial v} > 0$ and it appears that minimizing each criterion yields to the global minimization of the criterion.

(iv) Considering now forward links, let us consider at time t_0 the subset $\mathbf{v}_{\bullet_0} = (\cdots \mathbf{v}_n, \cdots)$ of cortical maps values which do not receive any forward connection, i.e. which are only connected to inputs. This defines a sub-lattice without forward connections. The related criterion $\mathcal{L}_{\bullet_0}$ is thus strictly decreasing and reaches its minimum, thanks to what as being discussed previously. Let us now consider the subset $\mathbf{v}_{\bullet_\tau} = (\cdots \mathbf{v}_n, \cdots)$ of cortical map values which do not receive any forward connection, except from $\mathbf{v}_{\bullet_{\tau-1}}$. Because $\mathcal{L}_{\bullet_{\tau-1}}$ is already minimized, forward links do not influence the related criterion $\mathcal{L}_{\bullet_\tau}$ which is thus strictly decreasing and reaches its minimum, By induction it appears that $\mathcal{L}_\bullet$ is minimized.