Available online at www.sciencedirect.com
ScienceDirect
Journal of Physiology - Paris xxx (2007) xxx–xxx
Journal of Physiology Paris
www.elsevier.com/locate/jphysparis

How do high-level specifications of the brain relate to variational approaches?

Thierry Viéville *, Sandrine Chemla, Pierre Kornprobst

INRIA, BP 93, 6902 Sophia, France

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). In this article, we describe how the variational framework, which encountered a large success in modeling computer vision tasks, has some interesting relationships, at a mesoscopic scale, with computational neuroscience. We focus on cortical map computations such that "what is to be done" can be represented as a variational approach, i.e., an optimization problem defined over a continuous functional space. In particular, generalizing some existing results, we show how a general variational approach can be solved by an analog neural network with a given architecture and conversely. Numerical experiments are provided as an illustration of this general framework, which is a promising framework for modeling macro-behaviors in computational neuroscience.

1. Introduction

Perceptual processes architecture, in computer or biological vision (Lee and Mumford, 2003; Burnod, 1993), is based on the computation of "maps" of quantitative values. The retinal image itself is a "retinotopic map": for each cell of the retina or each pixel of the image, there is a value corresponding to the image intensity at this location. This is a vectorial value for color images. A step further, in early-vision, the retinal image contrast is computed at each location, allowing to detect image edges related to boundaries between image areas. Such maps encode not only the contrast magnitude, but several other cues: 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. There are such detectors in both artificial visual systems (see e.g., Faugeras (1993) for a general introduction) and in the brain neuronal structures involved in vision perception (see e.g., Hubel (1994) for a classical overview). Such maps are not only parametrized by retinotopic locations, but also 3D locations, or parametrized by other parameters such as orientation, retinal velocity, etc. or more abstract quantities (Gisiger et al., 2000). Here, a map is going to be represented by a vector-value function from given domain (e.g., the retina surface) to a given range (e.g., the depth range).

In this context, the common point between computer or biological vision, is that both systems have to solve the same perceptual tasks and very likely make the same kind of hypotheses about the observed surroundings: they share the same internal representation. It is thus a relevant challenge to elaborate, at this level, a common theoretical framework in both fields, considering that the cortical computation is *specified* in a similar way although the models of implementation are obviously going to strongly differ.

At the biological level, cortical maps are well-defined, strongly architectured laminar structures of gray-matter interconnected via the white-matter. These maps can be scalar or vector-valued. This is an important feature when addressing the modelization of cortical processing units such as cortical columns (Burnod, 1993). It may also help

* Corresponding author. Tel.: +33 4 92 38 76 88.
E-mail address: thiery.vieville@sophia.inria.fr (T. Viéville).

0928-4257/$ - see front matter © 2007 Published by Elsevier Ltd.
doi:10.1016/j.jphysparis.2007.10.010

defining improved models of neurons or small neuronal assemblies, where the state is not only defined by a scalar membrane potential (Dayan and Abbott, 2001). These maps are also defined with non-linear constraints between components. This is an important feature in order to take into account the complex structure of a map (e.g. pin-wheel organization) via a non-linear differential structure (Petitot and Tondut, 1999). It is also useful to take noisy measures into account without any statistical bias (Viéville et al., 2001).

Computational neuro-scientists have a very profound view of the cortical maps architecture and how the neuronal micro and macro-circuitry allows the emergence of visual functions (Grossberg, 1988; Lee and Mumford, 2003; Friston, 2002; Dayan and Abbott, 2001). Such visual functions include edge detection, motion computation, segmentation (e.g. figure-ground segmentation or motion segmentation), focus of attention (on one visual token), etc. Both computer vision and studies of the visual system in computational neuroscience tackle such mechanisms. However when the computational models are re-used for artificial image processing on realistic inputs (not only for "toy" applications) the performances of the algorithms are often far from what is obtained by biologically non-plausible mechanisms.

Here, we obtain the following positive result: very powerful variational approaches can be implemented on, say, analog Wilson–Cowan-style recurrent networks, thus the apparent gap does not in-fact exist. It is just a matter of finding the suitable constraints on their connections. As a consequence, at the present state of our knowledge about cortical map processing, the main computer vision processing can be implemented using existing cortical map processing models. This is the main claim of this contribution. Let us detail how we can come to this point.

This paper is organized as follows. In Section 2 we review how the present variational approach is linked with high-level specifications of the brain perceptual processes. In Section 3, we introduce the notion of computational map and propose a variational definition, detailing its semantic and how it implements in analog networks at a mesoscopic scale. Finally Section 5 proposes several illustrations of the proposed framework.

2. Variational approaches in computational neuroscience

2.1. From generative models to generic estimation loop

2.1.1. Starting with representation learning

Following (Lee and Mumford, 2003; Friston, 2002; Dayan and Abbott, 2001), when considering high-level specifications of how the brain represents and categorizes its sensory input, we can start considering that the related cortical architecture is: *a machine to find "causes" v from inputs w*. In such a context, it has been proposed that the perceived external world can be viewed as a deterministic (or stochastic) dynamical system, e.g., of the form

$$\begin{cases} \dot{x}(t) = f(x(t), v), \\ w(t) = g(x(t)), \end{cases} \tag{1}$$

where the variable x is the "hidden" deterministic system state (usually a complex multi-variate non-linear quantity), with the initial condition $x(-\infty) = 0$.

In fact, using the so-called Fliess fundamental formula and related Volterra kernels (see, e.g., Dayan and Abbott, 2001 Chapter 10 for a review), we can eliminate the influence of x and directly relate the input to the recent history of the causes v:

$$w(t) = \underbrace{\int_0^t \kappa_1(\tau) v(t-\tau) \mathrm{d}\tau}_{\substack{\text{linear influence} \\ \text{from previous causes}}} + \underbrace{\int_0^t \int_0^t \kappa_2(\tau,\tau') v(t-\tau) v(t-\tau') \mathrm{d}\tau \mathrm{d}\tau'}_{\text{modulatory influence between causes}} + \cdots \tag{2}$$

including higher order terms, where the kernels κ_i are defined by

$$\beta = \left[\kappa_1(\tau) = \left.\frac{\partial w(t)}{\partial v(t-\tau)}\right|_{t=0}, \quad \kappa_2(\tau,\tau') = \left.\frac{\partial^2 w(t)}{\partial v(t-\tau)\partial v(t-\tau')}\right|_{t=0}, \ldots \right] \tag{3}$$

Thus have an equation of the form

$$w = P(v, \beta) \tag{4}$$

These abstract equations have a profound meaning. Dynamical system Eq. (1) can be considered as a universal way to represent a continuous system (here deterministic). Eqs. (2) and (3) state that providing that the "memory" is bounded, it is always possible to parameterize the whole class of representation (here with β). Thus, it is always possible to switch from the hidden state representation to a parametric representation. As a consequence predicting inputs from related causes amounts to estimate a parameter (of large dimension). It is hypothized that this is the way internal representation are learned in the brain. The related "learning" task would be equivalent to some "parametric learning". This key idea is in deep relation with the notion of adaptation (Holland, 1975; Viéville et al., 2001). The neural substrate for such mechanism of prediction and reward is not discussed here (see, e.g. Schultz et al., 1997; Sutton and Barto, 1998; Pasupathy and Miller, 2005). The link with "reward learning" is discussed in details in Friston (2002).

2.1.2. Yielding an estimation loop

Thanks to this previous formulation, estimating causes v from inputs u is a feed-forward and feed-back based process, which is described by several methods such as expectation–minimization (EM) method. The EM method has the two following stages (see Fig. 1):

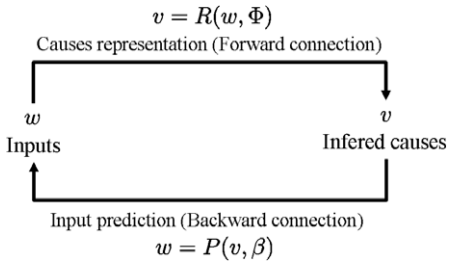

Fig. 1. Schematized view of the expectation–minimization (EM) loop. The inference being coherent if and only if: $w = P(R(w, \Phi), \beta)$. The internal representation of the causes is defined by function R and the internal representation of the external system by function P.

- Expectation, which "infers" the causes from the given inputs (here parametrized by forward connections Φ).
- Estimation, which "predicts" the input from *a priori* causes (here parametrized by backward connections β).

This abstract backward/forward duality shows that simply estimating the causes from the input would have been a completely ill-defined problem, whereas introducing this "estimation loop" allows the estimation mechanism to be well-defined and iteratively efficiently estimated (see, e.g., Rao and Ballard, 1999).

This very general architecture is probably the best simple view of cortical processing, assessed by the observation of the brain activity, at the present state of the art (Rao and Ballard, 1999; Ullman, 1995; Lee and Mumford, 2003; Friston, 2002; Dayan and Abbott, 2001) (see also in Friston (2002) a discussion about models based on the minimization of the mutual information in the system).

How can this general specification leads to an effective mechanism of estimation? The Bayesian framework is a popular choice.

2.1.3. Link with the Bayesian framework

The popular Bayes approach (Lee and Mumford, 2003; Friston, 2002; Dayan and Abbott, 2001) allows to instantiate the general architecture presented in Fig. 1, considering the "maximally probable" estimation (MAP) of the causes v, knowing w, thus:

$$\max_v \log(p(v|w)) = \max_v [\log(p(w|v)) + \log(p(v))] - \log(p(w)), \tag{5}$$

where $\log(p(w))$ is not to be considered because it is constant with respect to the quantity v to optimize. This maximal likelihood (5) optimization rewrites

$$\max_v \log(p(v|w)) = \max_v \log(p(w|v)) + \log(p(v)), \tag{6}$$

where the criterion is the sum of the conditional information and the *a priori* information. In fact, the first term is tuned by β so that $w = P(v, \beta)$, and the second term is tuned by Φ so that $v = R(w, \Phi)$. So (6) rewrites

$$\max_v \log(p(v|w)) = \max_v \log(p(P(v, \beta)|v)) + \log(p(R(w, \Phi))), \tag{7}$$

where each term is respectively the estimation and expectation. This is a canonical instantiation of the proposed architecture. In this context, the key step is to choose a probability model. The usual choice is to consider an additive random noise ε such that $w = P(v, \beta) + \varepsilon$, with an exponential probability distribution $p_\theta(\varepsilon) = e^{-V_\theta(\varepsilon)}$ parametrized by a vector-valued parameter θ. The probability distribution parameter θ is an *a priori* information, thus to be estimated during the expectation phase. This choice includes most distributions such as Gaussian, Poisson, binomial and uniform. More generally any Gibbs distribution related to this criterion can be used.

Finally, we have a dual forward/backward expectation/estimation criteria optimization:

$$\min_{\Phi,\theta} V_\theta(w - P(R(w, \Phi), \beta)), \tag{8}$$

$$\min_\beta V_\theta(w - P(v, \beta)). \tag{9}$$

More generally, this framework provides a quite profound highlight about important cortical mechanisms (e.g., contextual specialization, cues representation, etc.) as detailed in (Lee and Mumford, 2003; Lee, 2002; Rao, 2004).

2.1.4. Beyond the Bayesian framework

Two arguments are often raised against Bayesian formulations.

- One the one hand, certain aspects of the biological plausibility have been questioned. The fact, that no biologically plausible scheme for learning the required priors has ever been discovered. The fact that outside of sensory pattern recognition, Bayesian decisions often do not even roughly match those made by actual animals. This first indictment is discussed and partially answered in (Rao, 2004).
- On the other hand, it is emphasized that MAP-style regularization solutions are inadequate, and that rather something about the whole probability distribution of possible causes given the actual input, needs to be captured in neural population activity. This is because too many possibilities are viable given just the tiny-dimensional projection of the world in the input of which the MAP solution offers a very poor representation. This is discussed in Lee and Mumford (2003).

In fact, the right question is to know to which extends is it mandatory to introduce all *a priori* related to the Bayesian framework? If the answer is out of the scope of this paper, the present proposal has the advantage to be *compatible* with the Bayesian formulation, since the minimized criterion could be interpreted as a MAP estimation, but *without requiring* such a framework.

More precisely, the criterion to minimize can be directly specified, without any probabilistic interpretation. In

particular, the notion of prior simply corresponds to a term of regularization. Among all possible solutions, the idea is to select the most regular solution given a definition of regularity. A step further, the related probability distributions are parameterized thus reduced to a manageable space state.

Let us consider the expectation–estimation (EM) problem formalized in (8) and (9) for some cost function $V_\theta(\varepsilon)$ parametrized by θ, without any reference to a probabilistic formalism.

The expectation term in (8) corresponds to the *a priori* or internal knowledge, as discussed previously and the term in (9) corresponds to the input or external knowledge. In order to minimize simultaneously each term, we can minimize a linear combination of both. This constraints the set of possible solutions but the dual problem to solve is well-defined and a simple control on the balance between each term is available.

This is exactly what is used in a variational approach as detailed now.

2.2. Presentation of the variational framework: calculus of variations

Calculus of variations is a field of mathematics that deals with functionals, as opposed to ordinary calculus which deals with functions. A variational approach is defined by a minimization problem, where a functional $\mathscr{L}$ has to be minimized over a space of functions v defined on a continuous space denoted by Ω, and belonging to a functional space denoted here $H(\Omega)$ which defines its regularity. This can be written

$$\begin{aligned}\bar{\mathbf{v}} &= \underset{\mathbf{v}\in H(\Omega)}{\operatorname{argmin}}\, \mathscr{L}(\mathbf{v}) \\ &= \underset{\mathbf{v}\in H(\Omega)}{\operatorname{argmin}} \int_\Omega F(\omega, v(\omega), \nabla v(\omega), H(v)(\omega))\mathrm{d}\omega, \end{aligned} \tag{10}$$

where $\bar{\mathbf{v}}$ denotes the solution (when it exists). The integral in (10) shows the criterion to be minimized, which generally depends on the function v and its spatial derivatives. The standard way to estimate the solution $\bar{\mathbf{v}}$ is to derive and solve the so-called Euler–Lagrange equation, which corresponds to the gradient of the energy $\mathscr{L}$

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathbf{v}}(\bar{\mathbf{v}}) = 0.$$

Numerically, it is then classical to use a gradient descent approach for instance, so that one introduce a dynamical scheme and find $\bar{\mathbf{v}}$ as the steady state of

$$\frac{\partial \mathbf{v}}{\partial t} = -\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathbf{v}}(\mathbf{v}). \tag{11}$$

Eq. (11) is a partial differential equation (PDE), which relates the temporal evolution of v (left-hand side term) to its spatial derivatives (right-hand side term), i.e., values of the function v in a small neighborhood.

One of the main interests in using PDEs is that the theory behind the concept is well established. It means that functional analysis represents a wide literature which allows one to prove existence and uniqueness of a solution, but also its regularity. It is also our conviction that reasoning within a continuous framework makes the understanding of physical realities easier and stimulates the intuition necessary to propose new models.

Of course, PDEs are written in a continuous setting referring to analog images, and once the existence and the uniqueness have been proved, we need to discretize them in order to find a numerical solutions. There is a lot of flexibility regarding the way to discretize these PDEs. If finite differences are the most popular approaches in computer vision due to the regularity of the sampling, finite elements or finite volume methods can also be considered. In this article, we will show how to take into account a geometry of a sampling attached to a given network (see Section 4).

However, note that not all interesting visual processing can be specified by variational approaches. One may also define processes directly via PDEs which do not derive from any variational formulation. This framework is thus a very useful but not a universal approach to the low-level and middle-level visual processing.

2.2.1. Variational approaches in computer science

Indeed, the idea that cortical map computations can be related to variational approaches is not new. For example, it has been studied in low-level vision (Cottet and Ayyadi, 1998) which is at the origin of one aspect of this study. We also mentioned the link with high-level representation of cortical processing (Lee and Mumford, 2003; Friston, 2002; Dayan and Abbott, 2001), discussed in Section 1. For instance, the Mumford–Shah functional reviewed previously is well known in physiology (Carriero et al., 2003; Petitot, 2003) where it is used as a plausible model to account for image segmentation mechanisms at the perceptual level. Furthermore, a link has been proposed between neural oscillations in the cortex and the class of functional inspired from Mumford–Shah (Sarti et al., 2003).

A step further, the recurrent equation and related adaptation law of several neural networks models is derived from a functional, as detailed in (Likhovidov, 1997). It is shown in (Likhovidov, 1997) that the well known Grossberg networks (Grossberg, 1988; Raizada and Grossberg, 2003), counter-propagation networks, and Kohonen self-organizing networks (Fort and Pagés, 1994) are among them.

Interesting enough is the fact that, if the present paper focus on early-vision processes, variational approaches have been also used to formalized other cognitive functions. For instance, the fast-brain capability to categorize some ontology in 100–150 ms can be represented by a statistical learning criterion, the IT cortical map state corresponding to the optimum of this criterion (Viéville and Thorpe, 2004; Viéville and Crahay, 2004). Sensori-motor functions related to "trajectory generation" is efficiently represented by a so-called harmonic function potential

Please cite this article in press as: Viéville, T. et al., How do high-level specifications of the brain relate to ..., J. Physiol. (2007), doi:10.1016/j.jphysparis.2007.10.010

minimization (Viéville, 2006), this class of variational approach being also used to model some striatal functions (Connolly and Burns, 1993; Connolly et al., 2000) in relation with the generation of discrete and repetitive motions.

2.2.2. An efficient framework to model visual tasks

Traditionally applied in physics, variational approaches and methods based on PDEs have been successfully and widely transferred to computer vision over the last decade (see Aubert and Kornprobst, 2006 for a review).

We illustrate in Fig. 2 applications of this framework. The first line concerns image diffusion, which was historically at the origin of this success (isotropic smoothing, anisotropic smoothing Perona and Malik, 1990, effects Weickert, 1999). Notice the role of diffusion, done through linear or nonlinear local operators. Second example is image segmentation (Mumford, 1991), that will be commented further in Section 5.4. Segmented regions are shown here with random colors for display. Third example is inpainting which allows to reinvent a content based on local diffusion processes (Kornprobst and Aubert, 2006). Fourth example is flow restauration, to show that diffusion works for vector-valued images, with a flow that can be a local velocity estimation (optical flow) (Aubert et al., 1999; Tschumperle and Deriche, 2005). Last example is about sequence segmentation: Variational approaches are usually defined for still images but can be adapted for changing environments, i.e., sequence of images (Kornprobst et al., 2006).

Clearly computer vision addresses visual tasks similar to what has been observed in the visual system. Some examples include noise reduction as in the early-visual pathways, local edge detection as in V1, image completion (inpainting) as for the blind spot, motion detection as in MT, motion grouping and segmentation as in MST, image segmentation as in V2.

While modeling these different vision tasks, several general properties of variational approaches are revealed and can be related to cortical map computations. Main properties are:

- The process is specified as a criterion to minimize which is an *informative* way to define what is to process.
- Once the specification given, the related implementation is ''automatically'' derived from the Euler–Lagrange equation. This Euler–Lagrange equation is usually solved in dynamical process, defining a convergent process.

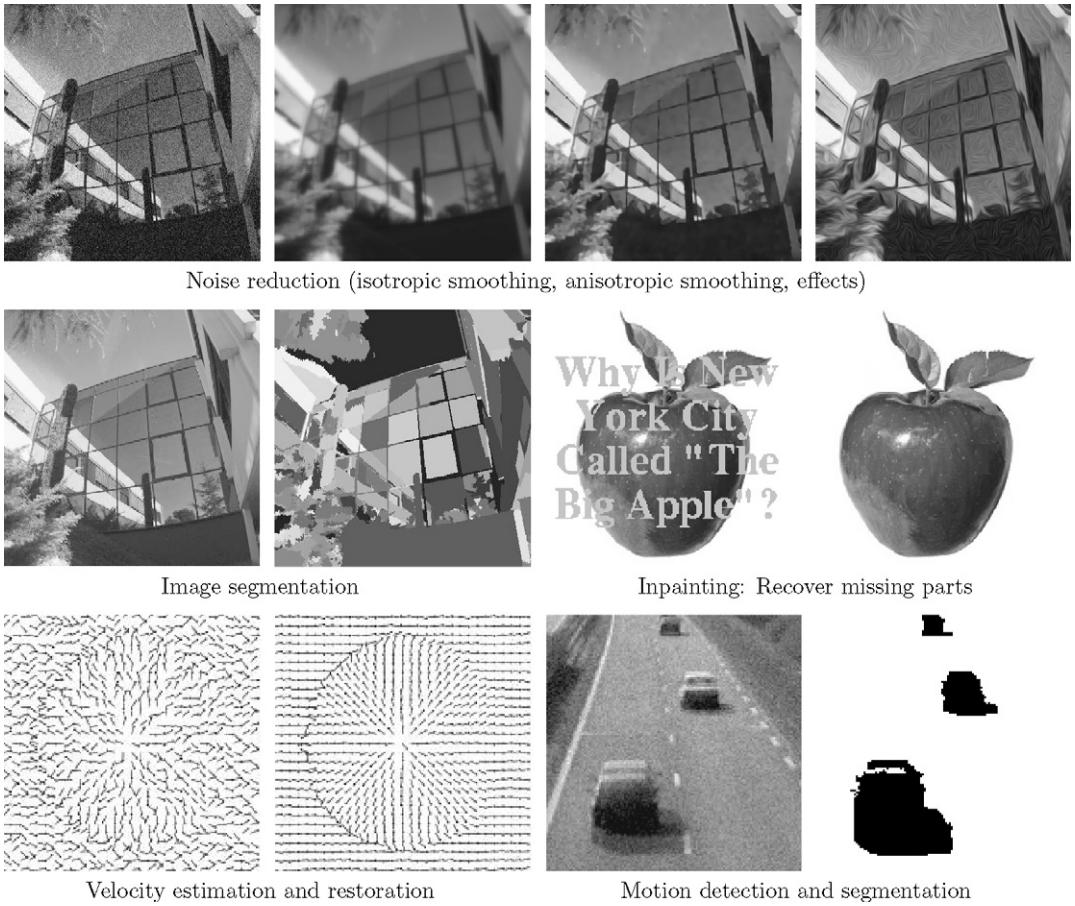

Fig. 2. Variational approaches in computer vision: a success story. From top to bottom, we illustrate some well-known applications in computer vision. See text for details.

Please cite this article in press as: Viéville, T. et al., How do high-level specifications of the brain relate to ..., J. Physiol. (2007), doi:10.1016/j.jphysparis.2007.10.010

- The process is highly parallel, defined by a set of distributed local processes, which cooperative calculations correspond to a global function.
- These local processes have two main ingredients: A local dynamic and a diffusion in a local neighborhood (think connectivity), exactly what happens in a neural network.
- The convergence is easy to control since we get a kind of "Lyapunov function" for free.
- The global result is obtained through a (generally non-isotropic and non-linear) input/output filter.
- Different scales can interact together.
- Finally, one can defined systems of coupled PDEs where several maps can interact together.

3. Specification of cortical map computation

3.1. General variational formulation

We consider that the goal of a neural map computation is to obtain an output map $\mathbf{v} : \mathbb{R}^n \to \mathbb{R}^p$, from an input map $\mathbf{w} : \mathbb{R}^n \to \mathbb{R}^q$. Note that vectors, and also matrices, are written in bold characters, matrices with capital letters and scalars in italic. For example, for $n = 2$, the space $\mathbb{R}^2$ could be a representation of the retina domain. If $\mathbf{w}$ is the retina monochromatic intensity, $q = 1$ (or $q = 3$ if color).

In this article, we focus on map computations which can be formalized as optimization problems. We propose here a very general variational formulation. Given an input map $\mathbf{w}$, one look for an output map $\bar{\mathbf{v}}$ verifying

$$\bar{\mathbf{v}} = \underset{\mathbf{v}\in H/\mathbf{c}(\mathbf{v})=0}{\operatorname{argmin}} \mathscr{L}(\mathbf{v}), \quad \text{with} \tag{12}$$

$$\mathscr{L}(\mathbf{v}) = \int_\Omega |\hat{\mathbf{w}} - \mathbf{w}|^2_{\mathbf{\Lambda}} + \int_\Omega \phi(|\nabla \mathbf{v}|_{\mathbf{L}}) + \int_\Omega \psi(\mathbf{v}), \tag{13}$$

$$\text{and } \hat{\mathbf{w}} = \mathbf{P}\mathbf{v}, \tag{14}$$

where ∇ stands for the gradient operator. $\mathbf{P}$, $\phi(\cdot)$, $\psi(\cdot)$, $\mathbf{c}(\cdot)$, $\mathbf{\Lambda}$ and $\mathbf{L}$ are commented hereafter. The norms defined in (13) are weighted norms defined by $|\mathbf{u}|_{\mathbf{M}} = \mathbf{u}^T\mathbf{M}\mathbf{u}$, where $\mathbf{M}$ is a given symmetric positive matrix. We also 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 $H = W^{1,p}(\mathbb{R}^n)$ (where p is related to the regularization term, i.e., the "shape" of function ϕ).

Let us interpret model (12)–(14) (see Fig. 3). The first term in (13) 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. Eq. (14) shows the chosen relation between the estimation of the input, given an output. So the formulation (12)–(14) 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.

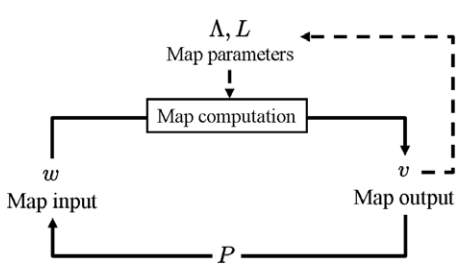

Fig. 3. The goal of a cortical map computation: obtain $\mathbf{v}$ from $\mathbf{w}$. In the sequel, we detail the different parameters drawn here.

We remark that model (12)–(14) is defined in a continuous framework. Of course, the ground truth of the neural units is discrete and we will show in Section 4 how to relate this global continuous formulation to a local discrete approach.

In the rest of this section, we comment some properties of the model (12)–(14).

3.2. Discussion about model features

3.2.1. Input control

The function $\mathbf{\Lambda} : \mathbb{R}^n \to S^+_m \in H$, where S^+_m is the set of square symmetric positive semi-definite matrices of size m, defines a so-called *measurement information metric*. It represents two properties:

- 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 $\mathbf{u}$, $\mathbf{\Lambda} = k\mathbf{u}\ \mathbf{u}^{\mathrm{T}}$ for some k), if the input is missing at one point we simply have to state $\mathbf{\Lambda} = 0$.

Very easily, writing $\mathbf{w} = \mathbf{H}^*\mathbf{w}'$ where $\mathbf{H}$ is any linear filter of the input $\mathbf{w}'$, the present formalism is also usable to estimate regularized version of any linear filtering (e.g., derivative to estimate a gradient or a velocity).

This corresponds to the "estimation" part of the EM formalism sketched out in Section 2.

3.2.2. Output regularization

The function $\mathbf{L}^{ij} : \mathbb{R}^n \to S_n \in H$, where S_n is the set of square symmetric matrices defines a *diffusion tensor* $\mathbf{L}$, which is symmetric (i.e. $\mathbf{L}^{ij} = \mathbf{L}^{ji}$) and positive (i.e. $\forall \mathbf{v} \in \mathbb{R}^n, \mathbf{v}^{\mathrm{T}}\mathbf{L}^{ij}\mathbf{v} \geqslant 0$).

The weighted norm of $\nabla\mathbf{v}$ is modulated by a function $\phi : \mathbb{R} \to \mathbb{R}$ 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 over-smoothed.

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 Aubert and Kornprobst, 2006 for a review and discussion). For

instance, a good choice is often the function $\phi(s) = \sqrt{1+s^2}$, convex and with linear growth at infinity.

When a problem is ill-posed, i.e. if there are many (and usually numerically unstable) solutions, adding some priors about 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 from well-defined values to undefined or ill-defined values.

This corresponds, with computational specification coming next, to the "expectation" part of the EM formalism sketched out in Section 2.

3.2.3. Computational specification

In order to further specify the computation, three kinds of constraints are introduced in (12)–(14):

- *Structural constraints* (written $\mathbf{c}(\mathbf{v}) = 0$), force the solution to 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 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. Viéville et al., 2001 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 relations* (written $\hat{\mathbf{w}} = \mathbf{P}\mathbf{v}$) between the input and the quantity to estimate. It is known that in order to obtained an unbiased estimation (see e.g. Viéville et al., 2001 for details) the measure itself has to be re-estimated or corrected. This corresponds to integrate $\hat{\mathbf{w}}$ in the estimation, thus in $\mathbf{v}$, as made explicit via the linear relation $\mathbf{P}$. This defines an internal feedback in the estimation process.

In this context, we consider $\mathbf{P}$, $\phi(\cdot)$, $\psi(\cdot)$, $\mathbf{c}(\cdot)$ as "fixed", whereas $\mathbf{\Lambda}$ and $\mathbf{L}$ are tunable. It would have been however easy to parametrized $\mathbf{c}(\cdot)$, $\psi(\cdot)$ and $\mathbf{P}$ with some additional parameters in order to allow more flexibility (e.g. a switch between two kinds of estimation). In fact, this appears to be useless in the present context: The simplest, the best.

One obvious question is how does a neuronal system "select" a given specification for a given cortical map. Here we propose to consider that the previous specification constraints correspond to what emerges with phylogenetic evolution, while the specification parameters are tuned by other computational maps (as in illustrated in the numerical section): Complex behaviors emerge from the interactions between computational maps.

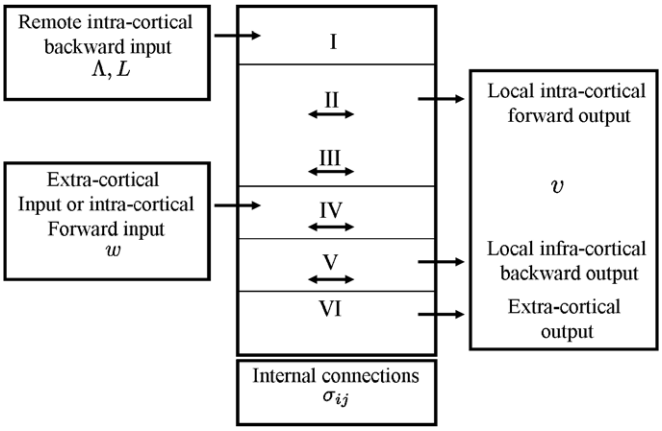

Fig. 4. Interpretation of the variables of the model with respect to a minimal representation of a cortical column connectivity. Extra-cortical input or intra-cortical input from previous layers, corresponds to $\mathbf{w}$, extra-cortical or intra-cortical output to $\mathbf{v}$. Local connections implement the diffusion operator parametrized by $\mathbf{L}$, while the input gain control, magnitude and geometry, is parametrized by $\mathbf{\Lambda}$. Remote backward connections allow to modulate $\mathbf{\Lambda}$ and $\mathbf{L}$.

3.3. The neural scale represented by the present model

In the context of the present study, such variational approaches allow to specify how a local neuronal unit contributes to the global computation issued by the cortical map. From the specification, the local state evolution and the diffusion between neuronal units are derived and well-defined.

In the cortex, such a "neuronal unit" is a cortical hypercolumn (see Burnod, 1993 for a treatise on the subject). Our model can be mapped onto usual computational model of cortical columns processes: regarding such a "processing unit", we propose in Fig. 4a possible interpretation of such an abstract analog network. This mapping makes explicit the scale at which such analog networks should be situated and has the chance to be compatible with the laminar architecture of the cortex or neocortex (Mumford, 1991; Douglas and Martin, 2004) and with the related inter-layer circuitry. However, this mapping is to be understood as a working assumption.

The complex geometric structure of cortical maps (i.e. pin-wheel organization) is known to correspond to non-linear differential structures of the cortical map (Petitot and Tondut, 1999). This is taken into account in our formalism, by the implicit equation $c(\mathbf{v}) = 0$ as made explicit previously.

4. Implementation on a neural network architecture

4.1. Network definition: sampling and connectivity

4.1.1. Sampling

If considering the micro-columns of a cortical map (Burnod, 1993), there is a clear sampling of the underlying continuous quantities. In Section 4 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 $\mathscr{S}_j$ around $\mathbf{y}_j$ defined by

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

where $\mu_j(\mathbf{y})$ is the measure density in $\mathscr{S}_j$. Here, we consider that $\mathscr{S}_j$ is bounded.

Two classical choices are $\mu_j(\mathbf{y}) = \delta(\mathbf{y} - \mathbf{y}_j)$ (Degond and Mas-Gallic, 1989) or $\mu_j(\mathbf{y}) = 1$ (Viéville, 2005b). The latter gives an average measure. Results presented hereafter do not depend on the choice of μ_j, which is a clear advantage since such a measure is never very specific in a biological model.

4.1.2. Connectivity

Let us consider a neural unit position $\mathbf{x} \in \mathbb{R}^n$, connected to a finite set of M samples $\{\mathbf{y}_1, \ldots, \mathbf{y}_j, \ldots, \mathbf{y}_M\}$ in a spatial neighborhood $\mathscr{S}$ of $\mathbf{x}$.

The word neighborhood is used here in his topological sense. At the biological level, $\mathscr{S}$ defines the receptive field of a sensory neuronal unit.

In order to model a biological neural network, it is important to consider very general connectivity patterns, as stated here. Furthermore, due to the huge complexity of the underlying mechanisms, we have to consider the weakest assumption about how each sample neighborhood.

More precisely, overlapped neighborhoods $\mathscr{S}_j \cap \mathscr{S}_i \neq \emptyset$ or partial partitioning $\cup_j \mathscr{S}_j \not\subseteq \mathscr{S}$ are allowed (see Fig. 5) and need to be taken account.

In order to relate the continuous specification defined previously to the discrete implementation formalized here, we must relate the discrete measures $\mathbf{v}(\mathbf{y}_j)$ to the underlying continuous quantity. In the derivation of the main result in (Viéville, 2005a) reported here, this link relies on the following *summation property* which defines a measure $\mu(\mathbf{y})$:

$$\begin{aligned}\mathbf{v}(\mathbf{x}) &= \int_{\mathscr{S}} \mathbf{v}(\mathbf{y})\mu(\mathbf{y})\mathrm{d}\mathbf{y} \\ &= \sum_j \int_{\mathscr{S}_j} \mathbf{v}(\mathbf{y})\mu_j(\mathbf{y})\mathrm{d}\mathbf{y} + \int_{\mathscr{S}-\cup_j\mathscr{S}_j} \mathbf{v}(\mathbf{y})\mu_\bullet(\mathbf{y})\mathrm{d}\mathbf{y}. \end{aligned} \quad (15)$$

Here $\mu_\bullet(\mathbf{y})$ is the measure density where no sample is available. This formula simply states that measures are related linearly, i.e., that the different samples are combined additively. It is verified by any sampling model we know (Degond and Mas-Gallic, 1989; Cottet and Ayyadi, 1998; Edwards, 1996; Viéville, 2005b), although often implicitly and not at this level of generality.

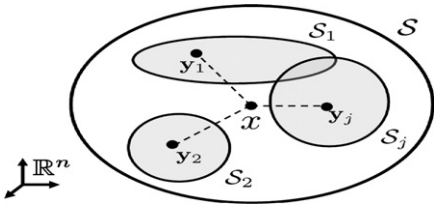

Fig. 5. Notations used to represent a sample neighborhoods with a general connectivity.

4.2. Relation between variational approach and neural network

4.2.1. Main proposition

Let us now make explicit the link between specification and implementation. We introduce the time t corresponding to the dynamics of the neural network state $\mathbf{v}(t)$.

Proposition 4.1. *The optimization problem (12)–(14) is, in the general case, locally minimized by the following linearized differential equation*

$$\frac{\partial \mathbf{v}_i}{\partial t} = -\varepsilon_i(\mathbf{v}_i) + \sum_j \sigma_{ij}(\mathbf{v}_i)\mathbf{v}_j + \kappa_i \mathbf{w}_i, \quad (16)$$

with

$$\begin{cases} \varepsilon_i(v) = \rho_i \mathbf{v} + \xi \frac{\partial \mathbf{c}}{\partial \mathbf{v}}^T \mathbf{c} + \frac{1}{2}\psi', \\ \rho_i = \sum_j \sigma_{ij} + \mathbf{P}^{\mathrm{T}}\mathbf{\Lambda}_i\mathbf{P}, \\ \kappa_i = \mathbf{P}^{\mathrm{T}}\mathbf{\Lambda}_i \end{cases} \quad (17)$$

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

$$\overline{\mathbf{L}} = \phi'(|\nabla \mathbf{v}|_{\mathbf{L}})\mathbf{L} \quad (18)$$

defined at M *points with*

$$M > \frac{(n+r)!}{n!r!} - \frac{n(n+1)}{2}. \quad (19)$$

They are given by solving the system

$$\begin{aligned} \overline{\mathbf{L}}_{\mathbf{kl}}(\mathbf{x}) &= \frac{\mathbf{1}}{\mathbf{2}} \sum_{\mathbf{j}} \sigma_{\mathbf{ij}} \bar{\mu}_{\mathbf{j}}^{\mathbf{e_k}+\mathbf{e_l}}(\mathbf{x}), \\ \mathbf{div}_{\mathbf{k}}(\overline{\mathbf{L}}(\mathbf{x}_\mathbf{i})) &= \sum_{\mathbf{j}} \sigma_{\mathbf{ij}} \bar{\mu}_{\mathbf{j}}^{\mathbf{e_k}}(\mathbf{x}), \end{aligned} \quad (20)$$

where[1]: $\bar{\mu}_j^\alpha(\mathbf{x}) = \int_{\mathscr{S}_j} (\mathbf{y} - \mathbf{x})^\alpha \mu_j(\mathbf{y})d\mathbf{y}$, *while they verify the following "unbiasness" conditions*

$$\forall i, \quad \sum_j \sigma_{ij}\bar{\mu}_j^\alpha(\mathbf{x}) = 0, \quad 2 < |\alpha| \leqslant r. \quad (21)$$

Among all σ_{ij} *verifying (20) and (21) we can choose those which verify*:

$$\min \sum_{ij} |\sigma_{ij}|^2. \quad (22)$$

The main message of Proposition 4.1 is to introduce Eq. (16) which corresponds to the standard Euler–Lagrange equation where the diffusion operator is discretized thanks to an integral approximation (even for vector-valued func-

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

tions). The second message, developed in the sequel, is to relate Wilson–Cowan networks defined by an equation of the form of (16) with the functional defined in (12) which is not only a specification but has also the role of Lyapunov functional.

The proof of Proposition 4.1 being quite technical, we preferred to put it in the end, and we refer to Appendix A for more details.

In the following sections, we comment the different equations of Proposition (4.1). Section 4.2.2 revisits the main part of this proposition which is how to approximate the diffusion operator **L** (Properties (18)–(20)). Section 4.2.3 presents the interest of this method on a simple example. In Section 4.2.4, we are also going to see that the network parameter calculation is fully automatic in this framework, with obvious engineering advantage. More than that, the fact that the specification/implementation mapping is one-to-one means that other cortical area can tune and feed-back on such cortical area, not only at the implementation level but also at the specification level.

4.2.2. Integral approximation of the diffusion operator

Integral approximation of differential operators have been introduced in the field of neural networks by Cottet and Mas-Gallic (1990), Cottet (1995), Cottet and Ayyadi (1998, 1996) and presented here in (Viéville, 2005a), where the derivation of the previous proposition is available. In particular, the present form provides an alternative to the use of so-called particles methods (e.g. Degond and Mas-Gallic, 1989) which neural interpretation is weaker.

The approximation of a differential operator by an integral operator is – in fact – mandatory since a differential operator is "punctual", whereas in the real life, it is not possible to apply an infinitesimally small operator. A step further, considering uncertainty, it is always relevant to relate an estimation not on one but on several values, i.e. define a non-punctual measure. This is exactly what an integral approximation, most often implicit, provides.

It is important to note, as demonstrated in (Degond and Mas-Gallic, 1989), that, due to the fact **L** is a positive operator, this integral approximation is not only closed to the related differential operator, but that it also leads to sampled solutions which are closed to the continuous solutions.

In the form of the previous proposition, these approximations are based on the summation property (15) and provide a direct link between a discrete integral approximation and the continuous differential operator, without the introduction of a "discretization" step. This is an improvement with respect to (Edwards, 1996; Degond and Mas-Gallic, 1989).

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

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

as shown in (Edwards, 1996), $\bar{\mu}_j^{\alpha}$ becomes arbitrary small so that *unbiasness* constraints (21) are automatically verified up to a negligible quantity when $|\alpha|$ increases. r is the order of the integral approximation unbiasness with respect to the relate differential operator.

Finally, the choice of (22) is "optimal" in the sense that the chosen integral approximation "as closed as possible" to the approximated operator in the least-square sense (see Viéville, 2005a for a complete discussion). In brief: the solution with the smallest "variance" is the closest to the punctual differential operator. In fact, not only the optimal approximation is defined by (22), but *a whole family of kernels*.

Thus, the linear sub-space defined by (20) and (21) implements a diffusion operator, including unbounded kernels (Degond and Mas-Gallic, 1989), allowing to represent a rather large class of networks as detailed in the next section.

4.2.3. Illustration of integral approximation precision

We would like to show the benefits of using Property 4.1 for discretization on a simple 2D example ($n=2$). Let us consider a classical restoration problem defined by the minimization of the functional

$$\mathscr{L}(\mathbf{v}) = \int_{\Omega} |\mathbf{v}-\mathbf{w}|^2 + \int_{\Omega} |\nabla \mathbf{v}|^2,$$

with a fidelity term and a quadratic regularization term (Tikhonov and Arsenin, 1977). In this simple case, L is the "identity matrix" (kronecker four order double symmetric tensor), and the Euler–Lagrange equation reduces to

$$\frac{\partial \mathbf{v}}{\partial \mathbf{t}} = \Delta \mathbf{v} - (\mathbf{v}-\mathbf{w}).$$

So here we have to discretize the Laplacian operator, which is usually discretized by a convolution with a 3×3 discrete mask like M_1:

$$\Delta v \approx M_1 * v \quad \text{where } M_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{23}$$

Now, let us show how the approach described in Proposition 4.1 (with a possible automatic implementation as described in Section 4.2.4) allows to obtain discretization with increased precision. In our approach, we need to introduce another parameter: the order r of the integral approximation, which also depends on the neighborhood size s (see Section 4.1). Thus, we can explore different values of both s and r, and we automatically generate the related masks.

Let us consider a given neighborhood size, defined by $s=2$, i.e., 5×5 masks since $M=(2s+1)^n$ for a simple neighborhood. Given r verifying (19), we can solve equations (18)–(20) to obtain some masks. For $r=2$ or 3 (respectively $r=4$ or 5), we obtain mask M_2 (respectively M_3) defined by

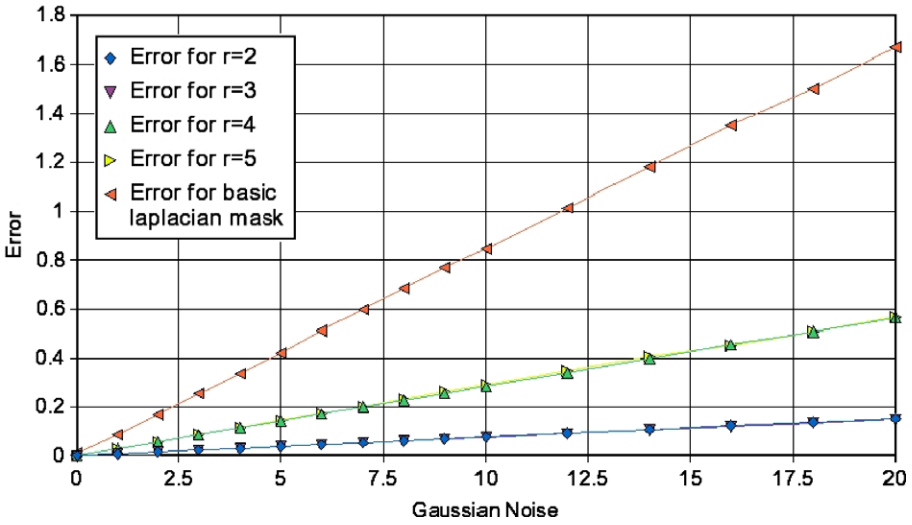

Fig. 6. Integral approximation order value optimization in the 2D case, for a regular image with squared masks and isotropic diffusion: Error between estimated discrete Laplacian and ground truth with different schemes and orders, with Gaussian noise.

$$M_2 = \begin{bmatrix} \frac{8}{135} & \frac{1}{27} & \frac{4}{135} & \frac{1}{27} & \frac{8}{135} \\ \frac{1}{27} & \frac{2}{135} & \frac{3473}{16230} & \frac{2}{135} & \frac{1}{27} \\ \frac{4}{135} & \frac{1}{135} & \frac{-20}{27} & \frac{1}{135} & \frac{4}{135} \\ \frac{1}{27} & \frac{2}{135} & \frac{3473}{16230} & \frac{2}{135} & \frac{1}{27} \\ \frac{8}{135} & \frac{1}{27} & \frac{-4771}{32460} & \frac{1}{27} & \frac{8}{135} \end{bmatrix},$$

$$M_3 = \begin{bmatrix} \frac{-1411}{16230} & \frac{257}{2164} & \frac{-4771}{32460} & \frac{257}{2164} & \frac{-1411}{16230} \\ \frac{257}{2164} & \frac{3578}{8115} & \frac{3473}{16230} & \frac{3578}{8115} & \frac{257}{2164} \\ \frac{-4771}{32460} & \frac{3473}{16230} & \frac{-1425}{541} & \frac{3473}{16230} & \frac{-4771}{32460} \\ \frac{257}{2164} & \frac{3578}{8115} & \frac{3473}{16230} & \frac{3578}{8115} & \frac{257}{2164} \\ \frac{-1411}{16230} & \frac{257}{2164} & \frac{-4771}{32460} & \frac{257}{2164} & \frac{-1411}{16230} \end{bmatrix}. \quad (24)$$

In order to compare the precision of these masks, let us consider an image for which ground truth is known (i.e., the Laplacian is known explicitly). We chose here the image of a Gaussian kernel. Fig. 6 shows the error curves between ground truth and estimated Laplacian with the different masks, as a function of a Gaussian noise to test robustness. Different values of r are shown. From this figure, we can observe that the masks computed with the previous formalism (24) give better approximations than the standard Laplacian mask (23), and we can also estimate the optimal value of the approximation order (for a given neighborhood $s = 2$) to be $r = 2$ here (corresponding to the smallest error curves in blue).

Similarly, considering the same problem with different s (see Chemla, 2006), optimal order r can be estimated. Results are shown in Table 1. Note that the maximal order is estimated thanks to inequality (19), s being fixed. Note that the maximal order is far from being optimal, and to given any other operator, deriving this optimal value is an open issue.

Table 1
Optimal and maximal order r computation in function of the smallest neighborhood size s

Smallest size s	1	2	3	4	5
Optimal order r	2	2	2	4	4
Maximal order r	3	5	8	11	14

4.2.4. Implementation of the network

Since the present integral approximation is obtained from a quadratic minimization (22) with linear constraints (20) and (21), it is well-defined and the solution is a closed-form linear function of $\overline{\mathbf{L}}$ and $\mathbf{div}(\overline{\mathbf{L}})$. Furthermore, this linear function is only defined by the network sampling, because only function of $\bar{\mu}_j^{\alpha}(\mathbf{x}_i)$.

Since $\overline{\mathbf{L}}$ is also a function of $\mathbf{v}$, σ is re-adjusted at each step. This is the reason why we write $\sigma_{ij}(\mathbf{v}_i)$ in (16). The fact σ is finally defined as a linear function of $\overline{\mathbf{L}}$ is essential at this stage. Furthermore, this corresponds to a linearized scheme and the convergence is obtained by an iterative mechanism. It also appears that the coefficients σ_{ij} can easily derived from a Hebbian rule (see e.g. Viéville (2005a) for a review) since they are defined by a quadratic criterion (22) with linear constraints (20).

As a consequence, the network parameters defined in (17) are directly given in closed-form from the specification equations $\mathbf{P}$, $\mathbf{c}(\cdot)$, $\psi(\cdot)$ and parameters $\mathbf{\Lambda}$ and $\mathbf{L}$, and all parameters are "compilable" in the following sense: Given any algebraic expression for the specification parameters, the implementation parameters are obtain by finite combination of parameter's derivatives and other closed-form symbolic derivations (see Chemla, 2006 for the development of this part of the study).

As made explicit in (Viéville, 2005a), the coefficient ξ must be *small* enough to decrease $\mathscr{L}$ and *high* enough to maintain $\mathbf{c}(\mathbf{v}) = 0$, which is obvious to adjust numerically, avoiding the explicit computation of the related formula. This is easy to control numerically. When non-linear constraints are not considered, $\xi = 0$.

Thanks to effective derivations we are able to "compile" the specifications, i.e. derive all neural network parameters in order to realize a computation. Clearly, the computational mechanism is not a "program" but is more likely related to a "iterative non-linear filter".

At a numerical level, we must simulate the continuous dynamics in a computer thus provide a temporal discretization of the dynamics. Not straightforward, but easy to formalize, please refer to (Chemla, 2006) for details.

4.3. Representation of analog network

4.3.1. Qualitative interpretation of Proposition 4.1

There is something more to learn from the previous result. If we consider an analog network defined in (16) with (17), we can detail and interpret each term:

- κ_i is the input gain control. In our framework it is directly proportional to the input reliability $\mathbf{\Lambda}$ and to the input–output relationship $\mathbf{P}$. At the implementation level, both terms are combined in (17).
- σ_{ij} are the neural network weights. In our framework they are in direct correspondence with the output regularization parametrized by $\mathbf{L}$. There is a one to one correspondence if we consider (22). Otherwise, many combination of weights implement a given regularization process as soon as (20) and (21) are verified up to an order r. This very large degree of freedom allows to hypothesize that a large set of diffusive neural networks could be represented in the present framework.
- ε_i defines the neural unit dynamics and contains two terms.
  - A leakage term parametrized by ρ_i which simply balances the terms related to the input gain control and the neural network weights.
  - Two corrective terms related to the constraints to verify, if any.

In other words, if the dynamics is defined by a leakage term which does not strictly corresponds to the input gain and network weights balance, it simply corresponds to the introduction of some weak constraint on the neural state. This again allows to account for a large range of neural networks.

4.3.2. Reciprocal result: when does a neural network solve a variational approach?

That we can make explicit the links between the neural network parameters and the variational specification is an interesting fact. Let us now formalize this fact and detail to which extends we can relate so-called Wilson–Cowan-style recurrent networks to a criterion of the form (12)–(14). Let us write Sig() the sigmoid function.

Proposition 4.2. *Given a network dynamics of the form*

$$\frac{\partial \mathbf{u}_i}{\partial t} = -\varepsilon_i(\mathbf{u}_i) + \sum_j \sigma_{ij}(\mathbf{u}_i)\mathbf{v}_j + \kappa_i \mathbf{w}_i, \tag{25}$$

with $\mathbf{v}_i = \mathrm{Sig}(\mathbf{u}_i)$, *as soon as the weights* σ_{ij} *are unbiased* (*i.e. verify* (21)), *then* (25) *locally minimizes in the general case the criterion*

$$\int_\Omega |\hat{\mathbf{w}} - \mathbf{w}|^2 + \int_\Omega |\nabla \mathbf{v}|^2_{\mathbf{L}} + \int_\Omega \psi(\mathbf{v}), \tag{26}$$

with $\hat{\mathbf{w}} = \kappa^{\mathrm{T}}\mathbf{v}$, $\psi = \int \varepsilon - [\sum_j \sigma_j + \kappa\kappa^{\mathrm{T}}]\mathbf{v}$ *and* $\mathbf{L}$ *defined by* (20).

This result is applicable to analog Hopfield network (as detailed e.g., in Edwards (1996)) and to the very powerful class of models Cohen and Grossberg dynamical system (e.g., Grossberg, 1988; Raizada and Grossberg, 2003) for which it has been shown (Viéville, 2005a) that the previous result is applicable. This has a very interesting consequence, since (26) has the same role as a Lyapounov functional: Since (25) minimizes (26) convergence of the related dynamical system towards a fixed point is guarantied. This property is verified for a very large class of weights σ since only (21) has to be verified. For instance, weights do not need to be symmetric.

A step further, a very large class of neural unit dynamics, defined by the non-linear term ε is compatible with this proposition: In fact, as soon as $\int\varepsilon$ is well-defined. There is a unique global solution if $\int\varepsilon$ is convex. Otherwise the local solution which depends upon the initial conditions.

Here a sigmoid non-linearity 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). This is important since it is a model of the relation between the membrane potential and the firing rate probability. At a technical level, our result is in fact true with or without this non-linear term (Viéville, 2005a). The reason is simple to understand: In the partial differential equation, after derivation, the non-linear term appears only as global gain factor in the minimization process. In other words, the non-linear relationship between u and v does not modify the variational criterion minimum, but only the rate of convergence. Since there is a saturation effect, convergence is slowed down for extreme values, which has very likely a stabilization effect.

One add-on of this specification is that convergence of such a network is demonstrated without the restrictive assumption of symmetry of the weights σ_{ij} (e.g. Grossberg, 1988).

Remark. This representation is well-defined for neural network connectivity with short-range connections, since this connectivity implements a local diffusion operator, and unbiasness conditions are numerically easily verified with

such kind of connections. Network mechanisms based on remote connections are, in the general case, not correctly represented in this framework, unless they correspond to unbiased weights σ, as defined (21). This is the case if remote connections implement diffusion at a large scale, or if a remote connection corresponds to some iterative short-range diffusion, integrated in one step in the remote link.

Since what is captured here is "anisotropic diffusion", this representation is well-defined for diffusive neural network connectivity. There is however no obstacle to have some of the weights with negative values (that the diffusion operator is positive does not means all coefficients are positive, see the mask M_3 in (24)). As a comparison, note that in the cortex a minority of weights are negative (about 20%, see (Burnod, 1993)).

5. Applications of proposed framework

5.1. Edge-preserving smoothing with contrast gain control in feed-back

Let us revisit an edge-preserving smoothing approach proposed by Cottet and Ayyadi (1998) which corresponds to the framework presented in this paper. In Cottet and Ayyadi (1998), given an initial image $w : \mathbb{R}^2 \to \mathbb{R}$, 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 $l = 1/\mathrm{Sig}'^{-1}$ (Sig being the sigmoid function), which is in fact related to the minimization of the criterion

$$\bar{v} = \underset{v}{\operatorname{argmin}}\, \mathscr{L}(v) = \lambda \int_\Omega (w - v)^2 + \int_\Omega |\nabla v|^2_{\mathbf{L}}, \tag{27}$$

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

$$\mathbf{L} = \left[\rho^2 \mathbf{P}_{\mathbf{g}^\perp} + \frac{3}{2}(1-\rho^2)\mathbf{I}\right] \quad \text{with} \quad \begin{cases} \mathbf{g} = S * \nabla v, \\ \rho = \min\left(1, \frac{|\mathbf{g}|^2}{s^2}\right), \\ \mathbf{P}_{g^\perp} = \begin{pmatrix} g_2^2 & -g_1 g_2 \\ -g_1 g_2 & g_1^2 \end{pmatrix}, \end{cases} \tag{28}$$

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 regularization term infers two complementary 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. 6 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 (28) corresponds to a Hebbian rule at the implementation level (Cottet and Ayyadi, 1998), and it can interpreted as a feed-back link from previous estimation of v onto the forward diffusion process (see the dotted arrow in Fig. 3).

It has been formally shown (Cottet and Ayyadi, 1998) that combining short-term adaptation with the diffusion process is a convergent process: The key point is that the feed-back from v to $\mathbf{L}$ is smooth in space, but not necessarily in time.

Contrary to (Cottet and Ayyadi, 1998), we have not introduced a non-linearity as discussed for (25): We have implemented a linear neural network as in (16). In the numerical simulations, we have verified that with or without this non-linearity, results are similar (up to the numerical precision), as expected since it does not modify the final minimal state to reach.

At step ahead, in (Cottet and Ayyadi, 1998), a temporal filtering is introduced in the feed-back. Thus, it is not directly $\mathbf{L}$ but an exponential temporal filtering of $\mathbf{L}$ which is taken into account. Our 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 ($0 \cdots 10$ times the sampling period) low-pass filter does not significantly influence the result, whereas higher delays inhibit the feed-back, inducing a convergence with only a poor edge-preserving smoothing.

Results are given in Fig. 7. The first example is the same as in (Cottet and Ayyadi, 1998) to validate the present method.

5.2. Iso-luminance estimation using constrained estimation

Let us revisit isotropic filtering, while a structural constraint $\mathbf{c}(\mathbf{v}) = 0$. More precisely, we choose to introduce the following constraint, for a red (r), green (g), blue (b) color image:

$$\mathbf{c}(\mathbf{v}) = r + g + b - \text{constant} = 0. \tag{29}$$

Fig. 8 shows the results on some of the previous images. The image is filtered but after the iso-luminance constraint is verified.

Furthermore, the convergence is very fast (only one iteration to verify the constraint and few iterations for the iso-luminance smoothing) since the constraint is linear. A step further, we applied non-linear constraint, like $r^2 + g^2 + b^2 - 255^2 = 0$, and have obtained similar convergence speed.

5.3. A winner-take-all mechanism using weak constraints

Let us now describe how a winner-take-all (WTA) mechanism can be written in this framework. WTA mech-

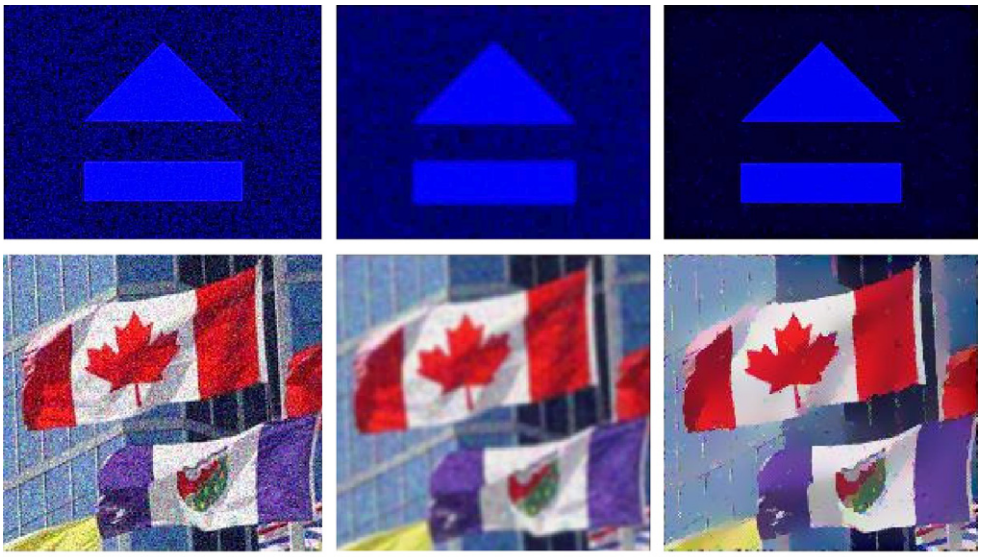

Fig. 7. Edge-preserving smoothing. The original image is on the left. As a comparison, a Gaussian filtering (isotropic diffusion) is shown in the middle. The synthetic image contains a huge (80%) amount of noise. The real image contains features at several scales. In both cases edges are preserved, while an important smoothing has been introduced.

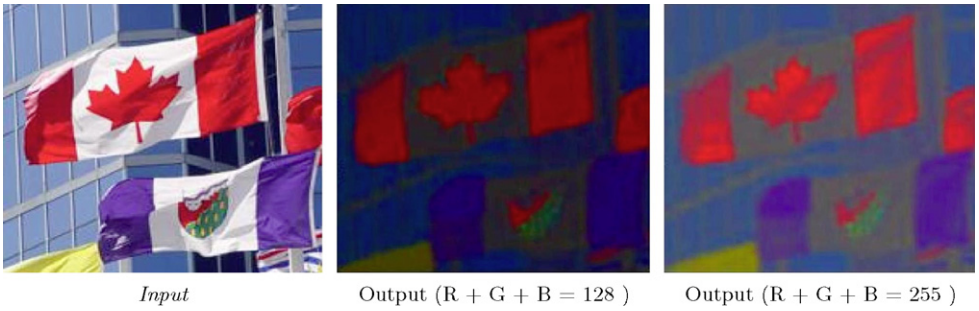

Fig. 8. Color image filtering with iso-luminance introduced as a structural constraint in the process. In the case of iso-luminance, the blue part remains almost perfectly blue and the dark black part becomes a gray-level part.

anisms are usually realized (e.g. Yu et al., 2003; Frezza-Buet et al., 2001; Grossberg, 1988) using an ad hoc mechanism with an explicit definition of inter-neuron inhibition in order to allow one neural unit to maintain its activity whereas all other activities vanish. They are used in many neuronal computations (see the review in Yu et al., 2003) and the way they could be implemented is still an issue. It is thus an 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

$$\bar{v} = \underset{v}{\operatorname{argmin}}\, \mathscr{L}(v) = \int_{\Omega} (w - v)^2 + \int_{\Omega} |\nabla v|^2 + \int_{\Omega} \psi(v), \quad (30)$$

where $\psi : [0,1] \to \mathbb{R}$ is a function, for example

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

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 characteristics: This non-linear term will force the values of the network to be zero or one, with a bias towards the zero value.

In Fig. 9 examples of result are shown. In order to avoid any parameter adjustment, 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 threshold adjustment policy is not critical. Results can also be obtained with a fixed threshold. 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 (Frezza-Buet et al., 2001;Grossberg, 1988), regarding WTA.

5.4. Implementation of a segmentation approach

Finally we report the implementation of the Mumford–Shah functional (Mumford and Shah, 1985), which is a

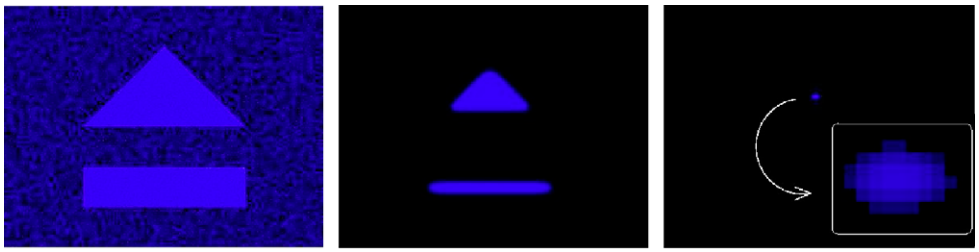

Fig. 9. Winner-take-all mechanism implemented using the proposed method. The very noisy (more than 80%) original image is on the left; the intermediate result shows how diffusion is combined with erosion yielding the final result, shown also with a zoom. Clearly the focus is given on the main structures of the image.

very classical problem in image processing. Results are proposed in Fig. 10 (see also Fig. 2).

The goal of the process is to segment an image defined on Ω into two or more regions of homogeneous intensity $W_1, W_2, \ldots$ separated by a set of border K, of total length $|K|$. Let us specify this wish in terms of a variational approach: We want to find the best segmented image v and the best region-border K with an optimal

- Fidelity: The output image v is as closed as possible to the input image w.
- Homogeneity: In each homogeneous region, the intensity variation is minimal.
- Parsimony: The length of the border should be minimal in order to keep only separations between very different regions.

Obviously, all these conditions cannot be fulfilled together and this is a matter of compromise which can be specified in the variational framework. This was proposed by Mumford and Shah (Mumford, 1991) who proposed the following variational approach:

$$\min_{v,K} \int_{\Omega} (w-v)^2 + \alpha \int_{W-K} |\nabla v|^2 + \beta |K|, \tag{31}$$

where $\beta > 0$ controls the fine/coarse grained segmentation and $\alpha > 0$ controls the scale of the segmentation providing an adjustable process with easily interpretable parameters.

There exists a wide literature on the theoretical study of (31) (see Aubert and Kornprobst, 2006), but we focus here on the implementation of this approach. The problem comes from the term $|K|$ which is difficult to derive.

We followed here the approach proposed by Ambrosio and Tortorelli (1990) where the set K is replaced by an auxiliary variable z (a function) that approximates the characteristic function $(1 - \chi_K)$, i.e., $z(x) \approx 0$ if $x \in K$ and $z(x) \approx 1$ otherwise. The authors proposed to minimize the following sequence of functional:

$$\mathscr{L}_{\varepsilon}(v,z) = \int_{\Omega} (v-w)^2 \mathrm{d}x + \int_{\Omega} z^2 |\nabla v|^2 \,\mathrm{d}x + \int_{\Omega} \left(\varepsilon |\nabla z|^2 + \frac{1}{4\varepsilon}(z-1)^2 \right) \mathrm{d}x,$$

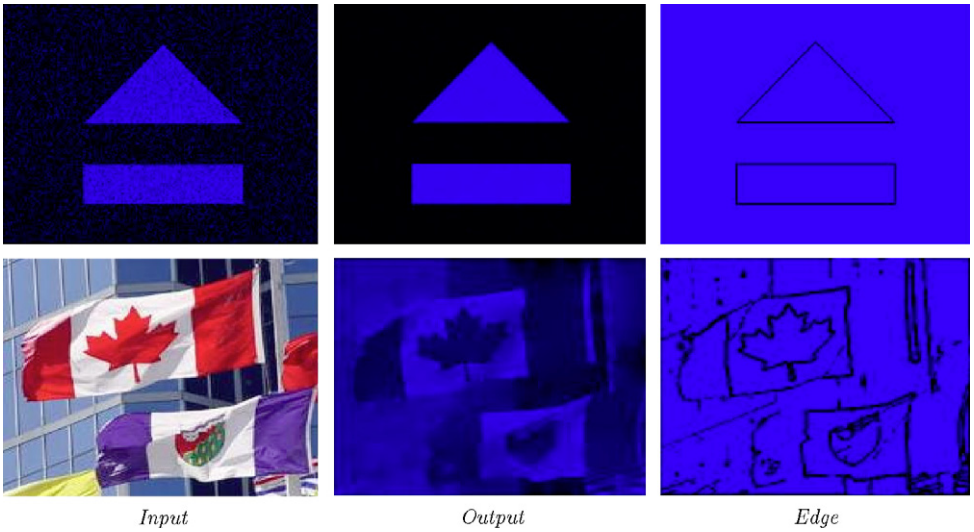

Fig. 10. Segmentation with the Mumford–Shah approach, based on the Ambrosio–Tortorelli approximation.

where ε controls the degree of admissible discontinuities thanks to compromise between the two terms in the integral. We refer to (Ambrosio and Tortorelli, 1990; Aubert and Kornprobst, 2006) for the precise study of this approximation, which is now under the form of our generic criterion (13) with vector-valued unknown.

6. 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 the related implementation on analog models.

Our goal was to show that the framework of variational approaches, so successful in modeling physics but also vision tasks in computer vision, had also some interest and relevance in computational neuroscience. Beside the proposed theoretical framework, we presented how qualitative properties of variational approaches match some properties encountered in local map computations. We also proposed some interpretations of our modeling in term of cortical columns.

More precisely we focused here on map computations that can be viewed as variational approaches, i.e., minimizing a global criterion, which has of course some relations with Lyapunov function. Our contribution was twofold. The first contribution was to generalize some results which relate variational approaches with a neural networks: Starting from a variational approach and a given geometry of the neural network, one can find the suitable synaptic weights which minimize the energy; Conversely, the energy associated to a given neural network can be defined under some assumptions. The second contribution was to investigate how to couple several variational approaches in a stable way. We proved that efficient and stable coupling could be obtained through some tuning parameters of the variational approach (namely the metrics involved in information measurement and diffusion). In this way, backward connections are "modulatory" for mediation of contextual effects.

We gave in this article several illustrations of the proposed framework, from vision task application to more fundamental processing such as winner take all mechanism. Mixing these applications will be the focus of our future developments, i.e., applications with map computations with feedbacks. Another focus will be to establish more connections between variational approaches and *spiking* neural networks.

In a nutshell, artificial networks are abstractions of biological entities and the variational approach allows to specify and compile them.

Acknowledgements

Powerful ideas of Jean Bullier are at the origin of this work. The reviewers help was precious. Lot of thanks to Olivier Rochel, Frédéric Alexandre and Laurent Perrinet for important discussions in relation with this work. This work is realized within the scope of the EC IP project FP6-015879, FACETS, in direct relation with scientific contributions of partners of this consortium.

Appendix A. Proof of Proposition 4.1

A.1. Deriving the Euler–Lagrange equations

Let us derive a differential equation

$$\frac{\partial \mathbf{v}}{\partial t} = F(\mathbf{v}),$$

so that $\mathbf{v}$ converges to the minimum of $\mathscr{L}(\mathbf{v})$ with the constraint $\mathbf{c}(\mathbf{v}) = 0$.

Writing $\mathscr{L} = \mathscr{L}(\mathbf{v})$, $\mathbf{c} = \mathbf{c}(\mathbf{v})$ and $\xi = |\mathbf{c}|^2$, the goal is to have

$$\frac{\partial \mathscr{L}}{\partial t} = \mathscr{L}_{\mathbf{v}} \frac{\partial \mathbf{v}}{\partial t} < 0 \quad \text{and} \quad \frac{\partial \xi}{\partial t} = \mathbf{c}^T \frac{\partial \mathbf{c}}{\partial \mathbf{v}} \frac{\partial \mathbf{v}}{\partial t} < -\varepsilon < 0, \tag{32}$$

where $\varepsilon > 0$ is a positive constant to be defined, and $\mathscr{L}_{\mathbf{v}}$ is the derivative of $\mathscr{L}$ w.r.t. $\mathbf{v}$ (i.e., the Euler–Lagrange equation without constraint). Here we assume $|\mathscr{L}_{\mathbf{v}}| > 0$, since no extremum is attained. Moreover:

$$\mathscr{L}_{\mathbf{v}} = 2\mathbf{P}^{\mathrm{T}}\mathbf{\Lambda}[\hat{\mathbf{w}} - \mathbf{w}] - 2\mathbf{div}(\phi'(|\nabla \mathbf{v}|^2_{\mathbf{L}})\mathbf{L}\nabla \mathbf{v}) + \frac{\partial \psi^{\mathrm{T}}}{\partial \mathbf{v}}, \tag{33}$$

which is obtained by standard calculus of variations (see Aubert and Kornprobst, 2006 for more details). Since both $\mathscr{L}$ and ξ are positive strictly decreasing quantities they will converge to a minimum. Moreover, since $\frac{\partial \xi}{\partial t} < -\varepsilon < 0$, ξ can not converge towards a strictly positive minimum and thus converges to 0 as required.

In order that $\frac{\partial \mathbf{v}}{\partial t}$ to verify (32), we propose the following equation:

$$\frac{\partial \mathbf{v}}{\partial t} \equiv -\mathscr{L}_{\mathbf{v}}^{\mathrm{T}} - \lambda \frac{|\mathscr{L}_{\mathbf{v}}^{\mathrm{T}}|}{|\frac{\partial \mathbf{c}^{\mathrm{T}}}{\partial \mathbf{v}}\mathbf{c}|} \frac{\partial \mathbf{c}^{\mathrm{T}}}{\partial \mathbf{v}}\mathbf{c}, \tag{34}$$

where $\lambda > 0$ is a constant to be defined in the sequel. Introducing the notation $\theta = \widehat{\mathscr{L}_{\mathbf{v}}^{\mathrm{T}}, \frac{\partial \mathbf{c}^{\mathrm{T}}}{\partial \mathbf{v}}\mathbf{c}}$, we can rewrite the expressions in (32) so that

$$\begin{aligned} &\frac{\partial \mathscr{L}}{\partial t} = -|\mathscr{L}_{\mathbf{v}}^{\mathrm{T}}|^2(1 + \lambda\cos(\theta)) < 0 \quad \text{and} \\ &\frac{\partial \xi}{\partial t} = -|\mathscr{L}_{\mathbf{v}}^{T}||\frac{\partial \mathbf{c}^{\mathrm{T}}}{\partial \mathbf{v}}\mathbf{c}|(\cos(\theta) + \lambda) < -\varepsilon < 0. \end{aligned} \tag{35}$$

If $|\mathbf{c}| = 0$ the constraint is verified. In (32), the right-hand-side condition is thus not to be considered, while the left-hand-side condition is verified for any λ in (34). Furthermore $|\frac{\partial \mathbf{c}}{\partial \mathbf{v}}| > 0$ since the constraints are regular.

Let us now consider the case where $|\mathbf{c}| > 0$. Conditions in (35) reduce to

$$1 + \lambda\cos(\theta) > 0 \quad \text{and} \quad \cos(\theta) + \lambda > \varepsilon'$$

for some $\varepsilon' > 0$ so that $|\mathscr{L}_{\mathbf{v}}^{\mathrm{T}}||\frac{\partial \mathbf{c}}{\partial \mathbf{v}}^T\mathbf{c}|\varepsilon' > \varepsilon$. If $\cos(\theta) > 0$ these conditions are verified for any positive $\lambda > \varepsilon' - \cos(\theta)$.

Please cite this article in press as: Viéville, T. et al., How do high-level specifications of the brain relate to ..., J. Physiol. (2007), doi:10.1016/j.jphysparis.2007.10.010

If $-1 < \cos(\theta) < 0$ (the case where $\cos(\theta) = -1$ is a singular case, the present solution being only valid in the general case) conditions are equivalent to:

$$\frac{-1}{\cos(\theta)} < \lambda < \varepsilon' - \cos(\theta)$$

and a solution exists as soon as $\varepsilon' < \frac{1-\cos(\theta)^2}{-\cos\theta}$. More precisely $\lambda > 1$.

The required conditions are thus verified in any case.

Now, considering the evolution defined in (34) in a continuous spatial domain, the goal is to show how it can be discretized in space, given a sampling and a connectivity pattern. From (34), we look for an approximation of the form of (16) and (17), which is directly derived from (33) and (34).

Note that in (16) it is sufficient to defined $\frac{\partial \mathbf{v}}{\partial t}$ up to a scale factor, which is a useful degree of freedom for time discretization.

A.2. Unbiased integral approximation of the differential operator

Clearly, the only difficult point is the regularization term derivation because the Euler Lagrange equations lead to the following well-known expression, which needs to be implemented in a discrete way:

$$\Delta_{\bar{\mathbf{L}}(\mathbf{x})}(\mathbf{f}(\mathbf{x})) = \mathbf{div}(\bar{\mathbf{L}}(\mathbf{x})\nabla\mathbf{f})(\mathbf{x}). \tag{36}$$

Here we freeze $\overline{\mathbf{L}}$, i.e. do not consider its dependence with respect to $\mathbf{v}$. The approximation is thus going to be valid for one step of the previous mechanism and iteratively adjusted with time.

Let us consider the following integral approximation:

$$\Delta^*_{\bar{\mathbf{L}}(\mathbf{x})}(\mathbf{f}(\mathbf{x})) = \int_{\mathscr{S}} \sigma(\mathbf{x},\mathbf{y})\mathbf{f}(\mathbf{y})\mathrm{d}\mathbf{y} - v(\mathbf{x})\mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{y}$$

for a kernel $\sigma(\mathbf{x},\mathbf{y})$ defined in $\mathbb{R}^n \times \mathbb{R}^n$, where $v(\mathbf{x}) = \int_{\mathscr{S}} \sigma(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{y}$.

This integral, in the discrete case where only $\mathbf{f}(\mathbf{y}_j)$, $l = 1,\ldots,M$ is available, cannot be estimated for a general kernel $\sigma(\mathbf{x},\mathbf{y})$, since on each input field $\mathscr{S}_j$, there is only one measure $\mathbf{f}(\mathbf{y}_j)$ and we thus can only define a unique weight $\sigma(\mathbf{x},\mathbf{y}_j)$. As a consequence, we choose a kernel of the form:

$$\sigma(\mathbf{x},\mathbf{y}) = \sum_{j/\mathbf{y}\in\mathscr{S}_j} \sigma(\mathbf{x},\mathbf{y}_j)\mu_j(\mathbf{y})$$

and obtain:

$$\int_{\mathscr{S}} \sigma(\mathbf{x},\mathbf{y})\mathbf{f}(\mathbf{y})\mathrm{d}\mathbf{y} = \sum_j \int_{\mathscr{S}_j} \sigma(\mathbf{x},\mathbf{y}_j)\mu_j(\mathbf{y})\mathbf{f}(\mathbf{y})\mathrm{d}\mathbf{y}.$$

This appears to be the more general kernel compatible with the present assumptions, yielding:

$$\begin{aligned}
\Delta^*_{\bar{\mathbf{L}}(\mathbf{x})}(\mathbf{f}(\mathbf{x})) &= \int_{\mathscr{S}} \sigma(\mathbf{x},\mathbf{y})\mathbf{f}(\mathbf{y})\mathrm{d}\mathbf{y} - v(\mathbf{x})\mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{y} \\
&= \sum_j \int_{\mathscr{S}_j} \sigma(\mathbf{x},\mathbf{y}_j)\mu_j(\mathbf{y})\mathbf{f}(\mathbf{y})\mathrm{d}\mathbf{y} - v(\mathbf{x})\mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{y} \\
&= \sum_j \sigma(\mathbf{x},\mathbf{y}_j)\int_{\mathscr{S}_j} \mu_j(\mathbf{y})\mathbf{f}(\mathbf{y})\mathrm{d}\mathbf{y} - v(\mathbf{x})\mathbf{f}(\mathbf{x})\mathrm{d}\mathbf{y} \\
&= \sum_j \sigma(\mathbf{x},\mathbf{y}_j)\mathbf{f}(\mathbf{y}_j) - v(\mathbf{x})\mathbf{f}(\mathbf{x}),
\end{aligned}$$

since $\mathbf{f}(\mathbf{y}_j) = \int_{\mathscr{S}_j} \mu_j(\mathbf{y})\mathbf{f}(\mathbf{y})\mathrm{d}\mathbf{y}$, while:

$$v(\mathbf{x}) = \int_{\mathscr{S}} \sigma(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{y} = \sum_j \sigma(\mathbf{x},\mathbf{y}_j)\int_{\mathscr{S}_j} \mu_j(\mathbf{y})\mathrm{d}\mathbf{y}.$$

This derivation gives the form of the integral operator in the discrete case. The goal is now the coefficients $\sigma(\mathbf{x},\mathbf{y}_j)$ of the discretization.

Let us now consider the Taylor expansion of $\mathbf{g}(\mathbf{y},\mathbf{x}) = \mathbf{f}(\mathbf{y}) - \mathbf{f}(\mathbf{x})$ with respect to $\mathbf{d} = \mathbf{y} - \mathbf{x}$

$$\begin{aligned}
\mathbf{g}(\mathbf{y},\mathbf{x}) &= \sum_{|\alpha|=1}^{r} \frac{\partial^\alpha \mathbf{f}}{\alpha!}(\mathbf{x})\bigg|_{\mathbf{y}=\mathbf{x}} (\mathbf{y}-\mathbf{x})^\alpha + o(|\mathbf{y}-\mathbf{x}|^r) \\
&= \left[\sum_{|\alpha|=1}^{r} (\mathbf{y}-\mathbf{x})^\alpha \frac{\partial^\alpha}{\alpha!}\right]\mathbf{f}(\mathbf{x}) + o(|\mathbf{y}-\mathbf{x}|^r).
\end{aligned}$$

We obtain:

$$\begin{aligned}
\Delta^*_{\bar{\mathbf{L}}(\mathbf{x})}(\mathbf{f}(\mathbf{x})) &= \int_{\mathscr{S}} \sigma(\mathbf{x},\mathbf{y})[\mathbf{f}(\mathbf{y}) - \mathbf{f}(\mathbf{x})]\mathrm{d}\mathbf{y} \\
&= \int_{\mathscr{S}} \sigma(\mathbf{x},\mathbf{y})\mathbf{g}(\mathbf{y},\mathbf{x})\mathrm{d}\mathbf{y} \\
&= \sum_j \int_{\mathscr{S}_j} \sigma(\mathbf{x},\mathbf{y}_j)\mu_j(\mathbf{y})\left[\sum_{|\alpha|=1}^{r} (\mathbf{y}-\mathbf{x})^\alpha \frac{\partial^\alpha}{\alpha!}\mathbf{f}(\mathbf{x})\right]\mathrm{d}\mathbf{y} \\
&\quad + \mathbf{R}^r\mathbf{f} \\
&= \left[\sum_{|\alpha|=1}^{r}\sum_j \sigma(\mathbf{x},\mathbf{y}_j)\int_{\mathscr{S}_j} (\mathbf{y}-\mathbf{x})^\alpha \mu_j(\mathbf{y})\mathrm{d}\mathbf{y}\frac{\partial^\alpha}{\alpha!}\right]\mathbf{f}(\mathbf{x}) \\
&\quad + \mathbf{R}^r\mathbf{f} \\
&= \left[\sum_{|\alpha|=1}^{r}\sum_j \sigma(\mathbf{x},\mathbf{y}_j)\bar{\mu}_j^\alpha(\mathbf{x})\frac{\partial^\alpha}{\alpha!}\right]\mathbf{f}(\mathbf{x}) + \mathbf{R}^r\mathbf{f},
\end{aligned}$$

where the remainder $R^\varepsilon_{kl}\mathbf{f}$ of this expansion may be written using an integral form:

$$\begin{aligned}
\mathbf{R}^r\mathbf{f} = \sum_{|\alpha|=r+1} \frac{r+1}{\alpha!}\int_{\mathscr{S}\times[0,1]} &\sigma(\mathbf{x},\mathbf{y})(\mathbf{y}-\mathbf{x})^\alpha(1-u)^r\partial^\alpha\mathbf{f}^j \\
&\times(\mathbf{x}+u(\mathbf{y}-\mathbf{x}))\mathrm{d}\mathbf{y}\,\mathrm{d}u
\end{aligned}$$

and because the support is bounded (as discussed in Section 4.1) thus included in a ball of radius ε, the remainder is bounded by the standard condition:

$$|R^r\mathbf{f}|_{0,\infty} < C\varepsilon^{r-1}|\mathbf{f}|_{r+1,\infty},$$

where C is a constant.

This would have been also true for an unbounded support providing $\sum_j \sigma(\mathbf{x},\mathbf{y}_j)\bar{\mu}_j^{\alpha}(\mathbf{x}) < +\infty, |\alpha| = r+1$ (see e.g. Degond and Mas-Gallic, 1989).

If we expand the diffusion operator with the same notations:

$$\begin{aligned}\Delta_{\overline{\mathbf{L}}(\mathbf{x})}(\mathbf{f}(\mathbf{x})) &= \mathbf{div}(\bar{\mathbf{L}}(\mathbf{x})(D\mathbf{f})(\mathbf{x})) \\ &= \mathbf{div}(\overline{\mathbf{L}}(\mathbf{x}))(\mathbf{Df})(\mathbf{x}) + \mathbf{trace}(\overline{\mathbf{L}}(\mathbf{x})(\mathbf{D^2 f})(\mathbf{x})) \\ &= \left[\sum_{\mathbf{k}} \mathbf{div}^{\mathbf{k}}(\overline{\mathbf{L}}(\mathbf{x}))\partial^{\mathbf{e_k}} + \sum_{\mathbf{kl}} \overline{\mathbf{L}}^{\mathbf{kl}}(\mathbf{x})\partial^{\mathbf{e_k}+\mathbf{e_l}}\right]\mathbf{f}(\mathbf{x})\end{aligned}$$

and identify with the previous expression of $\Delta^*_{\overline{\mathbf{L}}(\mathbf{x})}(\mathbf{f}(\mathbf{x}))$ up the rth order we obtain the conditions (20) and (21) for $|\alpha| > 0$.

We also finally obtain:

$$\Delta_{\overline{\mathbf{L}}(\mathbf{x})}(\mathbf{f}(\mathbf{x})) - \Delta^*_{\overline{\mathbf{L}}(\mathbf{x})}(\mathbf{f}(\mathbf{x})) = \mathbf{R}^r\mathbf{f}$$

leading to the proposed approximation.

References

Ambrosio, L., Tortorelli, V., 1990. Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. Communications on Pure and Applied Mathematics XLIII, 999–1036.

Aubert, G., Deriche, R., Kornprobst, P., 1999. Computing optical flow via variational techniques. SIAM Journal of Applied Mathematics 60 (1), 156–182.

Aubert, G., Kornprobst, P., 2006. Mathematics of image processing. In: Francoise, J.P., Nabar, G., Tosu, S. (Eds.), . In: Encyclopedia of Mathematical Physics, vol. 3. Elsevier, Oxford, pp. 1–9.

Burnod, Y., 1993. An Adaptive Neural Network: the Cerebral Cortex, 2nd ed. Masson, Paris.

Carriero, M., Leaci, A., Tomarelli, F., 2003. Calculus of variations and image segmentation. Journal of Physiology 97, 343–353.

Chemla, S., 2006. Biologically Plausible Computation Mechanisms in Cortical Areas. Master's thesis, Master STIC, Univ. Nice Sophia-Antipolis.

Connolly, C., Burns, J., 1993. A new striatal model and its relationship to basal ganglia diseases. Neuroscience Research 13, 271–274.

Connolly, C., Burns, J., Jog, M., 2000. A dynamical-systems model for parkinson's disease. Biological Cybernetics 83, 47–59.

Cottet, G., Mas-Gallic, S., 1990. A particle method to solve the Navier–Stokes system. Numer. Math., 57.

Cottet, G.-H., 1995. Neural networks: continuous approach and applications to image processing. Journal of Biological Systems, 3.

Cottet, G.-H., Ayyadi, M.E., 1998. A Volterra type model for image processing. IEEE Transactions on Image Processing 7 (3).

Dayan, P., Abbott, L.F., 2001. Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. MIT Press.

Degond, P., Mas-Gallic, S., 1989. The weighted particle method for convection–diffusion equations. Mathematics of Computation 53 (188), 485–525.

Douglas, R., Martin, K.A.C., 2004. Neuronal circuit of the neocortex. Annual Review of Neuroscience 27, 419.

Edwards, R., 1996. Approximation of neural network dynamics by reaction–diffusion equations. Mathematical Methods in the Applied Sciences 19, 651–677.

Faugeras, O., 1993. Three-Dimensional Computer Vision: a Geometric Viewpoint. MIT Press.

Fort, J.-C., Pagés, G., 1994. A non linear Kohonen algorithm. In: European Symposium on Artificial Neural Networks, Brussels, pp. 257–262, April.

Frezza-Buet, H., Rougier, N., Alexandre, F., 2001. Integration of biologically inspired temporal mechanisms into a cortical framework for sequence processing. In: Neural, Symbolic and Reinforcement Methods for Sequence Learning. Springer.

Friston, K., 2002. Functional integration and inference in the brain. Progress in Neurobiology 68, 113–143.

Gisiger, T., Dehaene, S., Changeux, J.P., 2000. Computational models of association cortex. Current Opinions in Neurobiology 10, 250–259.

Grossberg, S., 1988. Nonlinear neural networks: principles, mechanisms, and architectures. Neural Networks 1 (1), 1–97.

Holland, J.H., 1975. Adaptation in Naturel and Artificial Systems. Unpublished doctoral dissertation, university of Michigan.

Hubel, D., 1994. L'oeil, le cerveau et la vision: les étapes cérébrales du traitement visuel. Pour la science.

Kornprobst, P., Aubert, G., 2006. Explicit Reconstruction for Image Inpainting. Research Report No. 5905. INRIA.

Kornprobst, P., Deriche, R., Aubert, G., 2006. Image sequence analysis via partial differential equations. Journal of Mathematical Imaging and Vision 11 (1), 5–26.

Lee, T., Mumford, D., 2003. Hierarchical bayesian inference in the visual cortex. Journal of the Optical Society of America A 20 (7).

Lee, T.S., 2002. Top-down influence in early visual processing: a bayesian perspective. Physiology and Behavior 77, 645–650.

Likhovidov, V., 1997. Variational approach to unsupervised learning algorithms of neural networks. Neural Network 10 (2), 273–289.

Mumford, D., 1991. On the computational architecture of the neocortex. i. the role of the thalamo-cortical loop. Biological Cybernetics 65, 135–145.

Mumford, D., Shah, J., 1985. Boundary detection by minimizing functionals. Proceedings of the International Conference on Computer Vision and Pattern Recognition. IEEE, San Francisco, CA, pp. 22–26.

Pasupathy, A., Miller, E., 2005. Different time courses of learning-related activity in the prefrontal cortex and striaum. Nature 433, 873–876.

Perona, P., Malik, J., 1990. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7), 629–639.

Petitot, J., 2003. An introduction to the Mumford–Shah segmentation model. Journal of Physiology – Paris 97, 335–342.

Petitot, J., Tondut, Y., 1999. Vers une neuro-géométrie. fibrations corticales, structures de contact et contours subjectifs modaux. Mathématiques, Informatique et Sciences Humaines 145, 5–101.

Raizada, R., Grossberg, S., 2003. Towards a theory of the laminar architecture of the cerebral cortex: computational clues from the visual system. Cereb. Cortex 13, 100–113.

Rao, R., Ballard, D., 1999. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Natural Neuroscience 2 (1), 79–87.

Rao, R.P.N., 2004. Bayesian computation in recurrent neural circuits. Neural Computation 16 (1), 1–38.

Sarti, A., Citti, G., Manfredini, M., 2003. From neural oscillations to variational problems in the visual cortex. Journal of Physiology – Paris 97, 379–395.

Schultz, W., Dayan, P., Montague, R., 1997. A neural substrate of prediction and reward. Science 275, 1593–1599.

Sutton, R., Barto, A., 1998. Reinforcement Learning. MIT Press, Cambridge, MA.

Tikhonov, A., Arsenin, V., 1977. Solutions of Ill-Posed Problems. Winston and Sons, Washington, DC.

Tschumperle, D., Deriche, R., 2005. Vector-valued image regularization with pde's: a common framework for different applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (4), 506–517.

Please cite this article in press as: Viéville, T. et al., How do high-level specifications of the brain relate to ..., J. Physiol. (2007), doi:10.1016/j.jphysparis.2007.10.010

Ullman, S., 1995. Sequence seeking and counter streams a computational model for bidirectional information flow in the visual cortex. Cerebral Cortex 5, 1–11.

Viéville, T., 2005a. An abstract view of biological neural networks. Tec. Rep. No. RR-5657. INRIA.

Viéville, T., 2005b. An unbiased implementation of regularization mechanisms. Image and Vision Computing 23 (11), 981–998.

Viéville, T., 2006. About biologically plausible trajectory generators. In: International Joint Conference on Neural Networks, 2006.

Viéville, T., Crahay, S., 2004. Using an Hebbian learning rule for multi-class svm classifiers. Journal of Computational Neuroscience 17 (3), 271–287.

Viéville, T., Lingrand, D., Gaspard, F., 2001. Implementing a multi-model estimation method. The International Journal of Computer Vision 44 (1).

Viéville, T., Thorpe, S., 2004. A deterministic biologically plausible classifier. In 8th iccns. Boston University.

Weickert, J., 1999. Coherence-enhancing diffusion filtering. The International Journal of Computer Vision 31 (2/3), 111–127.

Yu, A.J., Giese, M., Poggio, T., 2003. Biophysiologically plausible implementations of maximum operation. Neural Computation 14 (12).

Please cite this article in press as: Viéville, T. et al., How do high-level specifications of the brain relate to ..., J. Physiol. (2007), doi:10.1016/j.jphysparis.2007.10.010