

# Neural networks: continuous approach and applications to image processing

Georges-Henri Cottet  
LMC-IMAG, Université Joseph Fourier  
BP 53x, 38041 Grenoble Cédex

December 30, 1997

## 1 Introduction

Our goal here is to describe elementary tools allowing to translate collective behavior of large neural networks in a continuous, with respect to time and space, framework.

Our starting point will be the Mac-Culloch and Pitts formal neuron, which operates in the following way: if  $(x_j)$  a set of neurons connected through a synaptic matrix  $J(x_i, x_j)$ , at every time step  $t$ , the states  $V_i^t$  of the neurons will be updated according to the rule

$$V_i^{t+1} = \begin{cases} 1 & \text{if } \sum_j J(x_i, x_j)V_j^t > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The important questions regarding the dynamical behavior of this very simple system are in particular:

- what is the asymptotic state, with respect to time, of the system
- how stable are this (or these) states
- how can boundary conditions affect the asymptotic states, in particular when the size of the network grows to infinity (phase transition problems).

Our motivation in looking for continuous alternative to neural networks is that we believe that continuous models can give new insights into these questions by highlighting the relevant parameters in the networks. In particular, continuous models can lead to new, and sometimes simpler, intuitions on the qualitative features of neural networks. Conversely, neural networks will also be seen as a natural basis for the construction of Partial Differential Equations for image processing.

Instead of the boolean dynamic (1) we will consider a time-continuous one suggested by Hopfield ([6]). In this model, besides the variable  $V$ , which now lies in  $[-1, +1]$  and denotes the output at the neuron locations, we introduce a state variable  $U$ , with value in  $\mathbf{R}$ , linked to  $V$  through the relation:

$$V = g(\lambda U) \quad (2)$$

where  $\lambda$  is a gain parameter. The states of the neurons evolve then according to the rule:

$$\frac{dU_i}{dt} = \frac{1}{n_i} \sum_j J_{ij}V_j - \alpha_i U_i \quad (3)$$

In this equation  $J_{ij}$  stands for  $J(x_i, x_j)$ ,  $\alpha_i$  are positive resistive coefficients and  $n_i$  is the number of non zero terms in the sum (in other words  $n_i$  is the number of neurons within the synaptic range of neuron  $i$ ).

Arguments from mean field theory can actually demonstrate that the system (2)-(3) is formally equivalent to the stochastic system obtained by replacing (1) by a probability transition law, a system often called the thermal version of (1). Our coefficient  $\lambda$  then plays the role of the inverse temperature  $T^{-1}$  and the function  $g$  is a generalization of the hyperbolic tangent generally introduced as a mollified heaviside function in these models.

For our purpose it will be enough to assume that  $g$  is increasing, odd and satisfies  $g(\pm\infty) = \pm 1$ .

The continuous systems we obtain are of reaction-diffusion type. This kind of system, which is very popular in various fields in biomodelling, are rather natural in view of (2)-(3): the right hand side of (3) contains a local averaging, and thus induces diffusion, while equation (2) enhances contrast just as a reaction term would do.

In section 2 we show how, and under which assumptions, it is possible to account precisely for these two ingredients to build the right reaction-diffusion equation. Then section 3 is devoted to the particular case of synaptic matrices obtained through Hopfield's algorithm. We show how to alter this algorithm in order to recover exactly, at least for the continuous model, any family of patterns that one wishes to store.

In section 4, we play the reverse game. Starting from a reaction-diffusion model involving a non linear anisotropic diffusion term designed to extract one dimensional objects inside a two dimensional image, we recover, at the discrete level, natural dynamical hebbian learning rules. This interpretation in turn suggests a new formulation of the reaction-diffusion system. The numerical examples we show in this section illustrate how discrete and continuous approaches can cooperate to provide efficient image processing tools.

## 2 Diffusion approximation

We are interested here in the approximation of *large* neural networks. By mapping them into a fixed box, it is clear that it is equivalent to consider networks where the distance between nearest neurons tends to zero. So we assume that the neurons are inside the unit box on a regular lattice:  $x_i = ih, i \in \mathbf{Z}^2$ .

To simplify we will first assume that the synaptic matrix  $J_{ij}$  is translation invariant and can be deduced from a compactly supported (with support in the unit ball for instance) function  $K$  with radial symmetry in the following way:

$$J_{ij} = K\left(\frac{x_i - x_j}{\varepsilon}\right)$$

The (small) parameter  $\varepsilon$  gives the synaptic range. Implicitely this means that we define on the network a topology where neighborhoods are deduced from the synaptic matrix. The number  $n_i$  in the right hand side of (3) is of the order of  $\varepsilon^2 h^{-2}$ . From now on we will assume that this number is constant throughout the network.

Our approximation will be valid in the limit of  $\varepsilon$  tending to 0 and  $n$  tending to infinity. If we keep in mind that we were starting from a network with side tending to infinity, this means that the distance between nearest neurons is small compared to the synaptic range, which itself is small compared to the size of the network.

The important parameter in the sequel will be the following momentum of the function  $K$

$$\tau_0 = \int K(y) dy; \tau_2 = \int y_1^2 K(y) dy = \int y_2^2 K(y) dy$$

Since the function  $g$  has been assumed to be increasing we can introduce its reciprocal function  $G$ , mapping  $] - 1, +1[$  onto  $\mathbf{R}$ . We then set

$$a(v) = \frac{\lambda\tau_2}{G'(v)} ; b(v) = \frac{1}{G'(v)}[\lambda\tau_0v - \alpha G(v)].$$

Our reaction-diffusion equation will be the following

$$\frac{\partial v}{\partial t} - \varepsilon^2 a(v)\Delta v = b(v) \quad (4)$$

Boundary conditions have also to be prescribed. The following results concerns periodic boundary conditions, but can be easily modified to handle other natural boundary conditions.

**Theorem 2.1** ([2]) *Let  $v$  the solution, assumed smooth enough, of (4), and  $V$  the solution to (2)-(3). Assume  $K$  smooth enough; then the following estimate holds*

$$\max_i |V_i(t) - v(x_i, t)| \leq \varepsilon^2 O(\varepsilon^2) + O(n^{-2})$$

The proof of this result follows from elementary arguments. The key observation is that, if  $v$  is smooth, its Taylor expansion combined with the symmetry properties of  $K$ , leads, up to an  $O(\varepsilon^2)$  correction, to

$$\tau_2 \Delta v(x) \simeq \varepsilon^{-4} \int K\left(\frac{y-x}{\varepsilon}\right)[v(y) - v(x)] dy = \varepsilon^{-4} \int K\left(\frac{y-x}{\varepsilon}\right)v(y) dy - \varepsilon^{-2}\tau_0v(x)$$

after a numerical quadrature of the above integral (using the neurons as integration points), we get

$$\tau_2 \Delta v(x_i) = \frac{\varepsilon^{-2}}{n} \sum_j J_{ij}v(x_j) - \varepsilon^{-2}\tau_0v(x) + O(\varepsilon^2) + \varepsilon^{-2}O(n^{-2}).$$

On the other hand, from (2)-(3) we easily deduce the following ODE:

$$\frac{dV_i}{dt} = \frac{\lambda}{G'(V_i)} \{n^{-1} \sum_j J_{ij}V_j - \alpha\lambda^{-1}G(V_i)\}.$$

Upon identifying the various terms entering the reaction function, we therefore conclude that  $V_i$  et  $v(x_i)$  satisfy, up to an  $\varepsilon^2 O(\varepsilon^2) + O(n^{-2})$  correction, the same equation, from which our result follows.

Concerning the reaction term  $b$  we make the following observation: one has  $b'(0) = \lambda\tau_0 - \alpha G'(0) > 0$  as soon as the gain parameter  $\lambda$  is large enough (in the analogy with thermal models, this means that the temperature is small enough, which is the interesting case)  $b(v)$  will have the sign of  $v$  and will vanish at 0 and for two opposite values  $v^*$  and  $-v^*$  in  $] - 1, +1[$ . We can therefore sketch the dynamical behavior of a solution in the following way: when  $\varepsilon$  decreases the trivial state  $v \equiv 0$  will lose its stability and, on a first time scale will give rise to states locally close to  $v^*$  or  $-v^*$ . The level sets corresponding to these 2 values will be separated by boundary layers. However, on a longer time scale, these fronts will then start moving with velocities typically proportional to their mean curvature, eventually leading to an asymptotic state where all traces of the initial state have disappeared.

Let us now consider two cases where the synaptic matrix will not be translation invariant, and thus will allow to reach more informative asymptotic states.

### 3 Hopfield's algorithm revisited

We are here interested in the following classical pattern recognition problem. Given a family of patterns  $\xi_i$ , we wish to construct a synaptic matrix such that these patterns will all be stable fixed points of the related dynamical system.

Hopfield's algorithm consists in using the following explicit form of  $J$

$$J(x, y) = \frac{1}{N} \sum_{i=1}^p \xi_i(x) \xi_i(y) \quad (5)$$

where  $N$  denotes the total number of neurons. It is readily seen that the states  $\xi_i$  will actually be fixed point of the iterations (1), provided these states are orthogonal to each other, in the sense of the discrete  $l^2$  scalar product on the network. In the general case, one can only prove weak versions of this result (namely the expected correlation between the asymptotic state and the stored patterns is close to 1 for a temperature large enough and in the limit of an infinite number of neurons, see [1])

Let us now analyze this scheme with the arguments developed in section 2. To start with, let us consider the case of  $p = 1$  pattern  $\xi$ , and let us alter Hopfield's algorithm by assuming that synaptic weights must keep a short range. We end up with the following function

$$J_\varepsilon(x; y) = \xi(x) \xi(y) \zeta\left(\frac{y-x}{\varepsilon}\right) \quad (6)$$

where  $\xi$  is a given smooth state and  $\zeta$  is a cut-off function with compact support. A Taylor expansion at order 4 of  $\xi$  around  $x$  yields

$$J_\varepsilon(x; y) = \xi(x) \left[ \xi(x) + (y-x) \cdot \nabla \xi(x) \right] \zeta\left(\frac{y-x}{\varepsilon}\right) + R(x, y)$$

In view of the analysis in section 2, to identify the diffusion resulting from this synaptic function, we need to evaluate, for a given smooth  $v$

$$\begin{aligned} \int J_\varepsilon(x; y) [v(y) - v(x)] dy = \\ \int \xi^2(x) \zeta\left(\frac{y-x}{\varepsilon}\right) [v(y) - v(x)] dy + \int \xi(x) (y-x) \cdot \nabla \xi(x) \zeta\left(\frac{y-x}{\varepsilon}\right) [v(y) - v(x)] dy \\ + \int R(x, y) [v(y) - v(x)] dy. \end{aligned}$$

If we now use a Taylor expansion of  $v$  at order 2 in the first integral, and at order 1 in the second, we obtain

$$\begin{aligned} \int \xi^2(x) \zeta\left(\frac{y-x}{\varepsilon}\right) [v(y) - v(x)] dy + \int \xi(x) (y-x) \cdot \nabla \xi(x) \zeta\left(\frac{y-x}{\varepsilon}\right) [v(y) - v(x)] dy \\ = \varepsilon^4 \tau_2 [\xi^2(x) \Delta v(x) + 2\xi(x) \nabla \xi(x) \cdot \nabla v(x)] + O(\varepsilon^6) \\ = \varepsilon^4 \tau_2 \operatorname{div}(\xi^2 \nabla v)(x) + O(\varepsilon^6). \end{aligned}$$

Similarly, we have

$$\int R(x, y) [v(y) - v(x)] dy = O(\varepsilon^6)$$

and thus we are led to a reaction-diffusion of the following type

$$\frac{\partial v}{\partial t} - \varepsilon^2 \operatorname{div}(\xi^2 \nabla v) = b(v) \quad (7)$$

Now we clearly see stable stationary states for this equation: these are the states whose values are either  $v^*$  or  $-v^*$  and such that  $\xi$  vanishes at the interfaces separating the corresponding level sets. If we start from a binary pattern  $\bar{\xi}$  that we wish to store this will happen if we take for  $\xi$  a regularization of  $\bar{\xi}$  which is zero at the interfaces.

If we now wish to store several patterns, the same analysis will apply, provided we start from the following rule

$$J(x; y) = \min_i \xi_i(x) \xi_i(y) \zeta\left(\frac{y-x}{\varepsilon}\right) \quad (8)$$

instead of (5): the reaction-diffusion model resulting from (8), will exhibit a diffusion coefficient  $\min_i \xi_i^2$  and thus will have the  $\bar{\xi}_i$  as stable stationary solutions. Note that this property is satisfied without any orthogonality or independence assumption for the patterns to be stored.

It is however important to notice that the validity of our diffusion approximation is restricted to *smooth* functions  $v$ , or at least functions which are smooth on a scale of order  $\varepsilon$ , so some care must be taken in the interpretation of this result at the discrete level. In particular when the network has only a small number of neurons, as it is often the case in practical applications, the notion of continuous approximation is itself probably meaningless.

## 4 Recovering a dynamical learning rule from non linear diffusion

We are now concerned with another aspect of neural networks. Rather than building *a priori* the synaptic matrix, we wish to allow the network to modify dynamically its connections in order to process the signal by a proper combination of filtering and contrast enhancement. In other words, we wish to keep the qualitative features that we recognized for the isotropic reaction-diffusion model on small time scales for longer time scales.

So-called hebbian learning rules can be proposed in this direction to mimic to some extent the real biological behavior of neurons, but they are not simple enough to allow a simple derivation of diffusion approximations.

This is the reason why we are going to proceed the other way: we will start from a "natural" non linear diffusion operator, then interpret it at the discrete level in terms of learning rules which will in turn suggest a modification of this operator.

Our diffusion operator will be of the form

$$Du = \operatorname{div}(L(\tilde{u})\nabla u)$$

where  $\tilde{u}$  denotes a perturbation of  $u$  that we will specify later, and  $L(\tilde{u})$  is the orthogonal projection on the direction  $\operatorname{grad} \tilde{u}^\perp$ . The motivation for considering this kind of operator has been to design filtering techniques which would not destroy one dimensional objects within a 2-D image (see [4])

In order to find an equivalent synaptic matrix, in the sense of the diffusion approximation performed in section 2, we need to write down an integral approximation of this differential operator. There is a general framework for doing this [5]. Integral operators have to be written in the following form

$$D_\varepsilon v = \int \sigma_\varepsilon(x, y)[v(y) - v(x)] dy$$

where

$$\sigma_\varepsilon(x, y) = \varepsilon^{-4} \sum_{i,j} m_{ij} \left(\frac{x+y}{2}\right) \psi_{ij}\left(\frac{y-x}{\varepsilon}\right)$$

The functions  $\psi_{ij}$  are cut-off functions related to the coefficients  $m_{ij}$  and to the diffusion matrix  $L$  by conditions that have to be imposed on their order 2 momentum. Our choice will follow one suggested in [5], with an alteration to make it fit better to our specific application. We take

$$\psi_{ij}(x) = x_i^\perp x_j^\perp \theta(x)$$

where  $\theta$  is a function with radial symmetry ( $x_i$  and  $x_j$  now denote the components of the vector  $x$  and neuron locations are denoted by indices  $p$  and  $q$ ). Upon a suitable normalization of this function, one can show that the matrix  $m$  leading to a consistent diffusion approximation of  $D$  reads

$$m = -L + \frac{3}{4}(\text{tr}L)Id$$

The synaptic matrix which results from this choice has the form

$$J_{pq} = \theta\left(\frac{x_p - x_q}{\varepsilon}\right) \sum_{i,j} m_{ij} \left(\frac{x_p + x_q}{2}\right) (x_p - x_q)_i^\perp (x_p - x_q)_j^\perp$$

It now remains to explicit  $m$  in function of  $L$ . We observe that, by the definition of  $L$

$$\sum_{i,j} L_{ij} \left(\frac{x_p + x_q}{2}\right) (x_p - x_q)_i^\perp (x_p - x_q)_j^\perp = \frac{1}{|\nabla \tilde{u}(\frac{x_p + x_q}{2})|^2} |\nabla \tilde{u}(\frac{x_p + x_q}{2}) \cdot (x_p - x_q)|^2.$$

Replacing the derivative of  $\tilde{u}$  in the direction of  $x_p - x_q$  by a centered finite difference (recall that due to the cut-off  $\theta$  only neurons  $x_q$  close to  $x_p$  matter), we obtain

$$J_{pq} \approx \theta\left(\frac{x_p - x_q}{\varepsilon}\right) |x_p - x_q|^2 \left[ \frac{3}{4} - \frac{1}{|\nabla \tilde{u}(\frac{x_p + x_q}{2})|^2} \frac{|\tilde{u}(x_p) - \tilde{u}(x_q)|^2}{|x_p - x_q|^2} \right]$$

One readily sees that, even if  $\theta$  is a non-negative function, the above expression can take negative values, depending on how much  $\tilde{u}$  varies between neurons  $x_p$  and  $x_q$ .

The above analysis indeed only makes it precise the intuition that if the diffusion is inhibited in a certain direction, at the discrete level this means that connections in this direction have to be weakened, possibly to reach negative values.

Let us now come back to our specific choice of  $\tilde{u}$  in terms of  $u$ . In [4] we used a regularization of  $u$  on a space scale of the order of a few neurons (or pixels in the image processing applications). It is then possible to prove that diffusion is strongly damped when the inputs present strong gradients along fronts whose curvature is not too large. This feature fits well with our primary goal which was to keep very thin objects. However in the light of the above analysis and if one believes that neural networks provide good guidelines in the derivation of efficient signal processing tools, there is another more natural choice, which is to compute  $\tilde{u}$  by means of temporal, rather than spatial, regularizations.

We then end up with a system of the form

$$\frac{\partial u}{\partial t} - \varepsilon^2 \text{div}(L \nabla u) = b(u) \quad (9)$$

$$\frac{dL}{dt} + \frac{1}{\alpha} L = \frac{1}{\alpha} \text{Proj}_{\nabla u^\perp} \quad (10)$$

with as initial condition on  $L$ , the identity (corresponding to an isotropic initial diffusion). This system now is the continuous analogue to a network where synaptic connections obey the law

$$\frac{dJ_{pq}}{dt} + \frac{1}{\alpha} J_{pq} = \frac{1}{\alpha} \theta\left(\frac{x_p - x_q}{\varepsilon}\right) |x_p - x_q|^2 \left[ \frac{3}{4} - \frac{1}{|\nabla u(\frac{x_p + x_q}{2})|^2} \frac{|u(x_p) - u(x_q)|^2}{|x_p - x_q|^2} \right]. \quad (11)$$

This is precisely the kind of hebbian rule we had in mind. The (small) parameter  $\alpha$  stands for the time scale on which learning takes place and the term between brackets induces either an enhancement or a penalty on the connections between neurons, depending on how synchronous these neurons are. It is however worthwhile to notice that, unlike in the hebbian models generally used, the activities of neurons are compared not in absolute value, but rather in balance of a mean correlation as measured by the gradient of  $u$ .

The mathematical and numerical properties of the model (9)-(10) are currently under investigation [3]. As a preliminary result we show in Figure 1 2 examples (already used in [4]) of medical images processed by this method. The left pictures show the original images. In the top right picture the tumor as well as the compressed part of the brain can be easily identified. The bottom images are obtained from a  $512 \times 512$  angiogram; the right picture, resulting from a quick (8 iterations) processing with a memory coefficient of 3 iterations, shows that vessels have been reasonably well extracted, and even reconnected, far away from the core, in areas where they are only a few pixels wide. Note that this image is obtained as the asymptotic of our model: performing more iterations will not change it significantly.

## 5 Conclusion

Our goal was to derive some elementary rules in order to translate discrete dynamical systems on neural networks into a continuous PDE framework and back. The latter framework renews the intuition one can have on the asymptotic states of these networks, in the limit of large networks, and conversely natural learning rules can motivate reaction-diffusion systems which can prove to be efficient in image processing. This approach also offers the possibility to use powerful PDE tools in the field of identification to allow the reconstruction of synaptic connections in the network from the observation of its dynamics.

## References

- [1] AMIT D., GUTFREUND H., SOMPOLINSKY H. - *Storing infinite numbers of patterns in a spin glass model of neural network*, Physical Review Letters, 55 (1985), 1530-1533.
- [2] COTTET G.-H.- *Modèles de réaction-diffusion pour des réseaux de neurones déterministes et stochastiques*, C. R. Acad. Sci. Paris, 312-I, 1991, pp217-221.
- [3] COTTET G.-H. and EL AYYADI M. - *Non linear PDE operators with memory terms for image processing*, in preparation.
- [4] COTTET G.-H. and GERMAIN L. - *Combining non-linear diffusion with reaction for image processing*, Math. Comp., 61 (1993), 659-673.
- [5] DEGOND P. and MAS-GALLIC S. - *The weighted particle method for convection-diffusion equations. II: the anisotropic case*, Math. Comp., 53 (1989), 485-508.
- [6] HOPFIELD J.J. - *Neurons with graded response have collective computational properties like those of two-states neurons*, Proc., Nat. Acad. Sci., 81 (1984), 3088-3092.

Figure 1: Image processing on medical images: top:  $256 \times 256$  MRI image of the brain;  
bottom:  $512 \times 512$  angiogram.