Are Neural Fields Suitable for Vector Quantization?

Lucian Alecu[(1,2)]
[1]CORTEX, Loria Nancy
615, rue du Jardin Botanique,
54600 Villers-lès-Nancy, France
Lucian.Alecu@Supelec.fr

Hervé Frezza-Buet[2]
[2]IMS, Supélec
2 rue Edouard Belin,
F-57070 Metz, France
Herve.Frezza-Buet@Supelec.fr

Abstract

This paper focuses on the possibility of enabling vector quantization learning techniques into dynamic neural fields, as an attempt to enrich their usage in bio-inspired applications. As mathematical approaches prove rather difficult to propose a practical solution, due to the non-linear character of the field equations, we adopt a different perspective in order to deal with this problem. This consists in simulating the evolution of the field and design an empirical method able to measure its quality. The developed benchmark framework implementing this approach is used to check whether a given field is capable to behave as expected in various situations, in particular those involving self-organization by vector quantization.

1 Introduction

In the field of machine learning, the so-called *unsupervised learning* techniques gather algorithms for which learning has to be performed without any clue about what has to be learned from data. In this context, learning implies analyzing the unknown density of probability that drives the production of the data set. In most cases, this analysis of the distribution means being able to represent the hidden density of probability in some compact way.

To do so, vector quantization algorithms use a finite set of *prototypes*, taken from the space where the data is defined. This set is the actual compact representation of the data mentioned above. The most famous algorithm is undoubtedly the generalized Llyod algorithm [7], also called LBG or k-means. This algorithm places k prototypes in the data space, so that the expectancy of the error made when considering a data as equivalent to its closest prototype is reduced. Since, three major refinements have been proposed to vector quantization. The first one is the introduction of *topology preservation*. This consists in linking prototypes so that they form a graph. The linkage is such that the induced graph topology fits the topology of the manifold underlying the density of probability that generates the data. This is achieved by a competitive procedure that produces an approximation of the Delaunay triangulation of the prototypes in the manifold [8]. The second refinement is a close idea, since it consists in setting a priori the linkage between prototypes, as a bi-dimensional grid for example. This is the core idea of Kohonen's self-organizing maps [6], that we will discuss further. Last, third refinement of vector quantization is the incremental approach [5], where the quantization accuracy is driven by adding or removing prototypes. The Growing Neural Gas (GNG) algorithm by Fritzke builds an increasing graph of prototypes until it "covers" the distribution of data.

Whatever the refinements used in some algorithms, we claim here that the very applicative power of vector quantization is the one regarding the ability to build a discrete structure (a graph, or a finite set of prototypes) from continuous distribution over some data space.

From a computational point of view, all the previously mentioned improvements of Llyod vector quantization rely on using a winner-take-most (WTM) competition process, instead of the winner-take-all (WTA) one. This has been used by early Kohonen works on SOMs [6], where references to the modelling of the cortical sheet were explicit. WTM consists in modifying more than one prototype when an example is given to the algorithm, and this has brought topology preservation to k-means paradigm, since the algorithm ensures that the winning prototypes form a small *contiguous* region in the graph. In early SOMs, each prototype is computed by an artificial neural unit, and finding the appropriate winning region relies on the relaxation of a dynamical process, based on on-center/off-surround influences among the units, a priori linked as a bi-dimensional grid. The dynamics of such bi-dimensional structure, called a *neural field*, has been studied for more than thirty years [1, 13, 3].

Neural fields provide a smart competition process, that has been used for robust selection of attention models in robotics [12, 11]. Nevertheless, learning, that should be coupled to such a competitive process, is hardly addressed in the literature. In efficient SOMs, a smart trick allows to implement the WTM procedure [6], by first choosing a single winning unit, and then applying a learning kernel around it (see equation 2). The aim of this paper is to identify why learning within competitive neural fields is so hard, whereas the Kohonen trick that models it works. Using neural fields in competition for vector quantization is crucial in some high level and massive self-organizing computational structures, that should run on parallel computer (Kohonen trick is sequential) and/or compute smart competition. This is the case for RFLISSOM model [10], that needs to be initialized by a SOM, whereas it attempts to address distributed learning for visual processing. The same remark stands for the bijama model [9] where the core investigated property is multimodal self organization. Some ad hoc field equations had to be used to keep the process distributed [4].

To sum up, vector quantization can be considered as a good candidate to bridge the gap between numerical and symbolic computation, as suggests cortical organization in humans [2]. In rich self-organizing architecture, Kohonen trick can not be used often, and designers need to return to the fundamentals of neural fields to set up a competition process that should drive the quantization of data flows. This challenge is not achieved yet, for reasons that are so obscure that this point has not been addressed in the literature as far as we know. This paper presents an original criterion that allows to qualify the performance of certain neural field equation. From this criterion, the reasons of the difficulties for allowing vector quantization in neural field-based computational modules are discussed with quantitative arguments.

2 An attempt to integrate learning mechanisms into dynamic neural fields

As previously mentioned, the number of models that use neural fields to accomplish learning tasks remains limited. Nevertheless, connectionist situated artificial intelligence applications would, at some point, implement both competition and learning mechanisms to solve complex tasks. These arguments motivated us to develop an in-depth study regarding the possibility of integrating vector quantization techniques in dynamic neural fields.

2.1 Case study

Dynamic neural fields are a mathematical formalism describing the activity of a population of neurons. More specifically, it consists of modelling the temporal evolution of the membrane potential, $u(x,t)$, for each neural unit x of a set of units, at any moment of time, t. The field is supposed to respond to the distribution of some other external activity, $i(x,t)$. Globally, the field response aims to choose the best responding units of the population (see figure 2). The $u(x,t)$ can therefore be viewed as the result of a competition driven by $i(x,t)$, enhancing the neural activity at relevant places in the field. The so-called *lateral* connections kernel weights within the field, typically having an on-center/off-surround shape, is involved in forming the response $u(x,t)$. The most commonly used neural fields are the ones described by S. Amari [1] as in the equation 1:

$$\tau \frac{\mathrm{d}u(x,t)}{\mathrm{d}t} = -u(x,t) + \int_{x'} w(|x-x'|) f(u(x',t)) \mathrm{d}x' + i(x,t) + h \quad (1)$$

The fact that each neuron in the field is competing with the others through local interaction kernel weights, makes this paradigm eligible to implement distributed and robust learning technique. Let us use it as the WTM mechanism in the classic Kohonen algorithm [6].

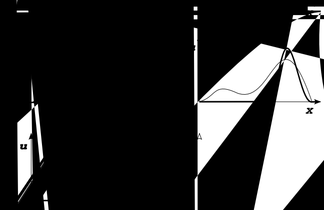
Briefly describing it, the algorithm aims to obtain a topographical mapping from the input data space (here $\Re^2$) onto a small dimensional array of nodes (here the field is composed by a “chain” of nodes). Each node i has an initially random reference vector m_i, called the weighting vector. In order to learn the topography of the input domain, random samples (two-dimensional vectors) are drawn from it, according to some hidden distribution $\mathcal{D}$ (the gray ring shown in figure 1). Each node evaluates then the matching between its reference vector and the one tossed from $\mathcal{D}$. Usually, this response, known as the *tuning curve*, is a decreasing function of the distance between the two vectors. Then, the location of the best matching node is computed. Finally, all the nodes within a neighborhood of the best matching node adjust their weights according to the following rule:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)], \quad (2)$$

where t is the current step of the algorithm, x is the given input sample and h_{ci} is the neighboring kernel, centered around the best matching node, allowing thus WTM instead of WTA. Usually, this one also is a decreasing function of the distance between the nodes within the network. This entire procedure induces a local relaxation on the weight vectors, leading in the end, after several steps, to their global ordering, and thus, to a topographical mapping of the input domain onto the array of nodes.

The above described Kohonen algorithm was implemented in two versions: the one just described above, and the other using neural fields stabilized states u as the neighboring kernel, instead of h_{ci} (see figure 1.b).

The vector quantization task supposes the network to reorganize in order to obtain a one-dimensional projection of the two-dimensional ring, which is actually obtained with



amp
his prove
d to achie
y shows th
to delive
a vec
ines
th
ur
D
s

mation from the input. In other words, the fact that the input is absent should not prevent the field from focusing activity somewhere. Rather than choosing among different options offered by the input, such a field decides to actually create information "out of nothing" in order to explore the environment, and not only to exploit it. Besides, we have already experimented this as a crucial feature in online learning architectures [9], since it allows the learning to start from scratch. A specific numerical instrument to capture and measure all these effects along the evolution of a neural field is described in section 3.3. Nevertheless, any other measurement can be used in the experimental framework, depending on the actual purpose of the neural field designer.

3.2 An empirical methodology for qualifying neural fields

The proposed methodology is implemented as a generic benchmark framework. A battery of predefined inputs composes the scenarios pool, while the above mentioned measuring tools account for the quantification instruments pool. This allows to quickly describe any benchmarking experiment simply by defining a scenario and a desired measuring instrument. Running multiple experiments would then give an empirical evaluation of the field average behavior in certain key situations.

3.3 A quantification instrument for neural fields

As described by the evaluation methodology, for each simulation step of an experiment, the field activity is quantified relative to a set of expected properties. As already stated, we consider neural fields as detectors of relevant regions of information within a given input: they grow so-called *bumps* of neural activity where the input is high. More formally, the following general property can be asserted:

(P): *Neural fields enhance their activity, u, where the input, i, is locally strong, ensuring that the distribution u evolves to a non-empty set of neural bumps.*

A quick review of the properties described in figure 2 would easily come to the conclusion that they are all the expression of (P). We consider thus the use of (P) as an implicit description of an optimality criterion both intuitive and practical, (P) being empirically perceived as a dominant behavior of an ideal field. Quantifying (P) would answer then *how well* this property is satisfied.

Reexamining the formulation of the (P) property, one could notice two particular features implied by it. First, stating that "u is enhanced where i is locally strong" infers that the output is locally fitting the input. Second, the neural activity is supposed to "enhance" and maintain high in order to detect relevant information from the input. Practically, this enhancement translates into generating uniform bumps of high amplitude within the field. In summary, the field is expected to stabilize to distributions of u *well-fitted* to the input (WFT condition) and having a *well-formed* shape of sparsely distributed bumps (WFM condition). We assume that the neural bumps in the field are neither too close, nor too far one from the others (sparsity constraint). The analysis that follows these remarks is not constrained by any bump shape. In this paper, we set the WFM condition as to "raise a single bump", which is in fact only a particular case of the sparsity constraint.

Let $\mathcal{B}$ define the family of all well-formed neural activity distributions. Therefore, $\mathcal{B}$ would comprise of all the possible ways of positioning the bumps across the field, respecting the sparsity constraint.

The following classical L^2 distance is further used to express the similarity between two distributions (e.g. the field input and the neural activity):

$$d(i,u) = \sqrt{\int_x ||i(x) - u(x)||^2 dx}$$

By abuse of notation, the distance between one distribution i and a set of distributions, $\mathcal{B}$, is defined as follows:

$$d(i,\mathcal{B}) = \min_{b \in \mathcal{B}} \{d(i,b)\}$$

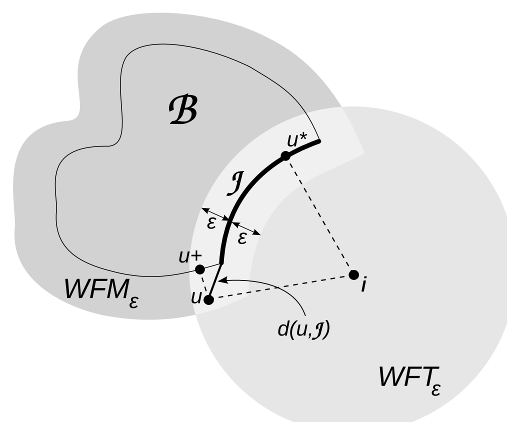


Figure 3. A geometrical interpretation of (P)

With the aid of these notations, the set of best-fitted elements of $\mathcal{B}$ to a given input i can be expressed as:

$$\mathcal{I} = \{u^* \in \mathcal{B} \,|\, d(i,u^*) = d(i,\mathcal{B})\}$$

The two conditions emerging from (P) are evaluated. The interest of the method is to measure how well (P) is satisfied by the input-output field distributions. As a consequence of the above analysis, satisfying (P) within an imposed ε margin implies satisfying both (WFT) and (WFM) within the same bound.

A well-fitted distribution u should not only minimize its distance to the input i, but also be not too far from the best-fitted ones from $\mathcal{B}$, i.e. $\mathcal{I}$. In other words, u should not fit i worse than any optimal distribution (WFT_ε) and it should also be close to a well-formed bump shape (WFM_ε):

$$d(i,u) \leq d(i,\mathcal{B}) + \varepsilon \qquad (\text{WFT}_\varepsilon)$$

$$d(u,\mathcal{B}) \leq \varepsilon \qquad (\text{WFM}_\epsilon)$$

Practically, in order to compute the distance from one particular distribution to the $\mathcal{B}$ set, an exhaustive research procedure is implemented.

Figure 4 provides the quality analysis picture of a given field, when the input i is kept constant until the state of the field stabilizes (e.g. for n steps in the figure). At each simulation step t, the satisfaction of the (P) property is evaluated numerically. Let $M_t(d(i,u), d(u,\mathcal{B}))$ designate the numeric expression of the two intrinsic conditions of (P) at the corresponding step t, and $M(d(i,\mathcal{B}), 0)$ the ideal satisfaction of (P) by the field. A qualitative field behavior should translate into a convergence of M_t towards M.

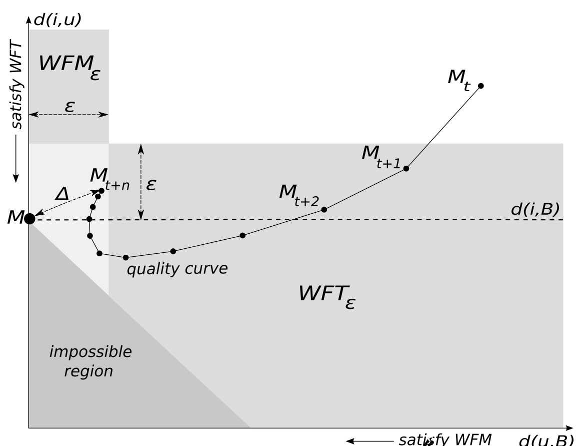


Figure 4. The plotting of (WFT) and (WFM) at different simulation steps gives the "quality curve" of the field.

It is worth mentioning that the procedure for evaluating (P) through its intrinsic effects is capable of tracking these two interrelated properties separately (as indicated by the two axes from the figure 4), enabling a way to perceive distinctly their influence upon the quality of (P). For example, one may argue that well-fitting is a much more significant feature of a qualitative field. Our proposed measuring method easily allows to track precisely this specific aspect, as proved by figure 4. The imposed conditions are sufficient to fully define a convergence criterion for (P). Moreover, since d is a distance, the following inequality thus stands from (WFT_ε) and (WFM_ε):

$$d(i,u) \geq d(i,\mathcal{B}) - \varepsilon$$

As a result, the impossible region in the figure 4 plot appears. Intuitively, it suggests that it is impossible for u to be very similar to i (thus very well-fitted) and also well-formed, and yet satisfy (P). This would hold only if i itself is close to $\mathcal{B}$. In particular, if i is an element of $\mathcal{B}$, the impossible region completely disappears.

4 Experimental results

The previous section studied a general property that we considered to define a qualitative behavior of the neural fields. In addition, a numerical procedure was proposed to measure it. Regardless the key features exhibited by this property, one may still argue that it was introduced as a hypothesis based on intuition rather than on biological data or theoretical formalisms. The current section proves that actually, this property is capable to make possible an efficient implementation of learning techniques in neural fields.

The same Kohonen algorithm described in section 2 was reimplemented in a third version of WTM, using as h_{ci} in equation 2 a bump that perfectly satisfies the (P) property. This is obtained by an exhaustive combinatorial research that makes sense only as a reference to evaluate the competition mechanism described by equation 1. Even if at the start of the algorithm the precision is low (caused by the randomness of the weights values, due to the inherent unlearned input samples), the best matching unit selection operation converges towards the global maximum one. The figure 1.c shows, as a consequence, that such so-called qualitative neural field behavior provides a way to arrive at very similar results as the classic Kohonen algorithm to perform the vector quantization task. This experiment indicates that the introduced (P) property is capable to enable competitive vector quantization using dynamic neural fields. The choice of describing the optimal states of the field (i.e. $\mathcal{B}$) is an a priori statement, but the described experiment proves that the assumed hypothesis regarding a qualitative behavior of the neural fields is worth taking into consideration and should be further exploited in researches concerning the performance of neural fields.

In order to deeply inspect the causes that determined the common neural field implementation to fail (figure 1.b), we used the above presented empirical methodology to design two more key experiments. In the first one (that we call E1), a single Gaussian of 1.0 amplitude was randomly positioned in the field every new 100 steps. In addition, random noise of maximum 0.4 amplitude was spread throughout the entire field. In the second experiment (E2), only the noise was kept. Quickly analysing them, one could easily find that the two experiments are describing the scenarios from figure 2. The first one aims to test the aggregated effects of a. and b., while E2 concerns the c. case from the mentioned figure. The Δ quality factor (see figure 4), defined

Experiment	$\overline{\Delta}$	σ_Δ
E1 (one Gaussian + low noise)	1.07504	0.439
E2 (only low noise)	2.12860	0.423
Kohonen (with Amari neural field equation)	2.16102	0.565
Kohonen (with qualitative neural field)	0	0

Table 1. Values of the average quality factor Δ for the same neural field equation benchmarked in different scenarios.

as the distance from a stabilized state of the field and an optimal one, was computed every time just before changing the input data (thus, only for the stabilized states). Low Δ values (ideally zero) indicate high quality behavior of the field. Evidently, the field was expected to emerge only one high bump of activity throughout the field. The location of the bump obviously corresponds to the location of the Gaussian in the first scenario, and to the most significant patch of input in the second one. Our inspected field equation succeeds in behaving well in the first experiment, while growing only very low amplitude single bumps in the second experiment. These verbal remarks are immediately proved by the quantitative measurements of the Δ factor, summarized in the table 1. As seen there, the average quality of the field behavior in E1 is significantly better then that recorded in E2. As a consequence of the bad performances of the field shown in the second experiment, one may quickly infer a low performance in the Kohonen algorithm implementation. Indeed, the numerical values (see also table 1) clearly indicates a high value of average Δ, meaning a low global performance of the field in such scenario. While the equation may be suitable for applications concerning attentional process simulation (given the fact that the field detects well noisy Gaussian patterns E1), it fails to be useful for implementing a vector quantization technique. In conclusion, the introduced empirical approach proves a useful tool to inspect various features of the neural fields in order to study their average behavior in different situations. As discussed, the proposed quantifying instrument is able to deliver such a global evaluation of the quality of the field.

5. Conclusions

The current paper focused on the possibility of integrating classic learning techniques into bio-inspired dynamic neural fields architectures. As an example, a simple vector quantization task was then defined as a suitable scenario to test the capacity of the field to implement such a technique. A generic empirical methodology to approach the study of various features concerning dynamic neural fields was proposed as an alternative to the analytical resolution of the equations governing their evolution. Using it, neural fields were qualified vis-à-vis a general formulated property, sensitive to an entire set of presumed key effects. Finally, simulation results showed that neural fields satisfying this property are good candidates for implementing the vector quantization technique. Given the benefit of the presented general framework aimed to empirically qualify the neural fields evolution in key scenarios, we intend to use this methodology in our future work to find a suitable neural field equation and adjust it in order to accomplish the goal of well satisfying (P). Thus, we will then be able to use it in the development of multimodal distributed self-organizing modules.

References

[1] S. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.

[2] Y. Burnod. *An Adaptive Neural Network: The Cerebral Cortex*. Masson, 1989.

[3] S. E. Folias and P. C. Bressloff. Breathers in two-dimensional neural media. *Physical Review Letters*, 95, 2005.

[4] H. Frezza-Buet and O. Ménard. Making competition in neural fields suitable for computational architectures. In W. Duch, J. Kacprzyk, and E. Oja, editors, *International Conference on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science (LNCS) 3696, part I, pages 217–222. Springer-Verlag, 2005.

[5] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, Cambridge MA, 1995.

[6] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, 1988.

[7] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–94, 1980.

[8] T. M. Martinez and K. J. Schulten. Topology representing networks. *Neural Networks*, 7(3):507–522, 1994.

[9] O. Ménard and H. Frezza-Buet. Model of multi-modal cortical processing: Coherent learning in self-organizing modules. *Neural Networks*, 18(5-6):646–655, 2005.

[10] R. Miikkulainen, J. A. Bednar, Y. Choe, and J. Sirosh. *Computational Maps in the Visual Cortex*. Springer, 2005.

[11] I. Mikhailova and C. Goerick. Conditions of activity bubble uniqueness in dynamic neural fields. *Biological Cybernetics*, 92(2):82–91, 2005.

[12] N. P. Rougier and J. Vitay. Emergence of attention within a neural population. *Neural Networks*, 19(5):573–581, 2006.

[13] J. G. Taylor. Neural networks for consciousness. *Neural Networks*, 10(7):1207–1225, 1997.