Journal of Network and Innovative Computing.
ISSN 2160-2174 Volume 8 (220) pp. 001–015
© MIR Labs, www.mirlabs.net/jnic/index.html

Collective brain surrogates

Bruno Apolloni [1] and Ernesto Damiani [2]

[1] Department of Computer Science
University of Milano
via Comelico 39/41 20135 Milano, Italy
apolloni@di.unimi.it

[2] Center on Cyber-Physical Systems
Khalifa University
P O Box 127788, Abu Dhabi, UAE
ernesto.damiani@ku.ac.ae

Abstract:
Within the framework of ensemble methods, we investigate on a *compatible* learning scheme, denoted as *learning by gossip*, with the aim of assessing its feasibility when facing a rather complex target function. Compatibility is expressed in terms of probability that the learned function could be actually at the basis of the observed training set, hence an explanation of it. Feasibility is in terms of the related Mean Square Error (MSE) on test sets. Elaborating on ways to improve the performance of the learning scheme, we assessed its reliability, efficacy and exploitability, via numerical tools that play the role of learning, educating, feeling and achieving consciousness in a virtual society. We base or conclusions on both theoretical and numerical arguments that are tossed on a well known benchmark. We devote a large space to provide graphical evidence to some conclusions that may be exploited in the field of both connected societies and cooperative computation frameworks.

Keywords: Collective brain, learning by gossip, compatible explanation, ensemble learning, subsymbolic kernels.

I. INTRODUCTION

In principle, two brains work differently because there are no synapses from the neurons of one brain the ones of the other. However, if you cut one brain in two you do not get two running brains. Mostly, this occurs because you break up the complex organization of neurons in a brain [1]. Taking in mind this drawback, we proceed in the opposite direction by grouping brains. This is a primordial attitude of human societies that lies at the basis of their progresses. In a very rough synopsis, surrogates of the missing inter-brains synapses have been, in a incremental way along the time: sub-symbolic messages like gestures, voiced words, written words, and again sub-symbolic messages like biosignals. Actually, subsimbolic signals re-gained a prominence with the evolution of the communication channels that, again from remote to recent times, have been: sensory channels like eyes and hears, paper, radio-telephone, e-mail and social networks. Science fiction prophetically anticipated an enormously powerful surrogate as for both contents and channel represented by telepathy (Isaac Asimov's Second Foundation [2]). The underlying assumption is that brains become able to deeply communicate one another so that the organization of each neural circuit is preserved and virtual synapses feed all signals needed for a powerful integration in an overall neural network, thus creating a *Collective Brain* (CB).
The closest analogy we can think of is an ensemble of brains linked via a social network. Brains may share documents, messages and sub-symbolic signals as well, the most popular being represented by the "I like" flag (the Boolean result of a non coded process). While sharing deep documents, like a philosophy book, among selected people may achieve a stronger brain plasticity effect fostering powerful minds, the mandates de-facto of current social networks are: 1) universality of the members and 2) simplicity of their interactions. They have both social and technological reasons and support a sort of society of minds pervading not only the political sphere but also the scientific one, hence of our interest.
In this paper we delve into this surrogate of collective brain, in view of evaluating how far is it from the original idea and, above all, what advantage it brings with respect to a single brain. Since we are computer scientists and not biologist we will pursue this goal by substituting, with all caveats, biological brains with artificial neural networks.
As for mandates, universality is straightforwardly realized in terms of scalability of our ensemble. Simplicity, instead, is a key issue. Simplification is a keyword of modern social life designating the general aim of removing superfluous rules and actions from customary interactions among people and between people and institutions. Concerns may arise when we transfer this philosophy to the interaction between people and natural phenomena, e.g. to scientific matters. Undoubtedly, facing complex phenomena such as those involving biological organisms or complex social systems, e.g. those related to traffic[3], we are compelled to simplify, model, and synthesize our observations in order to obtain suitable, though approximate, explanations. Using the paradigmatic framework of artificial neural networks, we adopt a first simplification consisting of abandoning the quest for a formal (symbolic) explanation of a phenomenon in favor of a suitable simulation of it. To this end, one may consider using a neural network to combine symbolic functions that locally *understand* the phenomenon under study. This approach is

known as the *hybrid ANN* paradigm [4]. In this paradigm, the neural network acts as *glue* connecting *knowledge parcels* of formal knowledge. Alternatively, we may replace the local symbolic functions with simple neural networks called *Gossiping Parcels* (GPs) to be combined with either majority voting or a decision tree on the parcel outputs when the overall explanation is a classification rule, or with a linear combination of the outputs in the case of a continuous function. All these schemes require a *learning phase* where the parcels and/or the glue are trained to simulate the phenomenon under study. This training can be regarded as *ensemble learning* [5, 6], i.e. using multiple learning algorithms to obtain better accuracy than the one that could be obtained from any of the constituent learning algorithms alone.

Our approach is to tackle this hard framework consisting of highly complex phenomena coupled with granular information within a general perspective of searching not the truth about these phenomena, rather explanations of them that are *compatible* with the observed data. Thus, we abandon the ambition of determining optimal strategies or simply describing phenomena through exact laws. Rather, we relieve the hardness of the exact solution by looking at looser ones that are *compatible* with our observations and our target, at least to a reasonable extent. This is not a minor goal, just in the province of practitioners; rather we may support its achievements with a well-founded theory that extends the bases of the Algorithmic Inference approach [7] developed in more canonical contexts. Hence, in this paper we recall the *learning by gossip approach* [8, 9], as a general umbrella for many ensemble methods, like ELM [10, 11] and Reservoir Computing [12], which perform parallel interconnection of simple learning units (called *gossipers* or *weak learners*, as they undergo only limited training). We focus on rigorously characterizing the gossiper's activity, in order to exploit the information they bring, in spite of their intrinsic roughness.

Learning by gossip, as an implementation of ensemble learning, extends the learning facility from a "divide et impera" perspective, which in modern acceptation could be called "Object Oriented Learning": in place of a monolithic model focused to solve the learning problem in toto, we fragment it in subproblems whose solutions are properly combined to get the final answers. The way of fragmenting are various, the common goal is to deal with weak learners to be combined in an optimal ensemble. The general idea is that while weak learners are simpler to train, though less accurate, their inaccuracy can be statistically recovered by the combiner (a perceptron, an SVM, a majority voter, etc.). Here, we focus on an ensemble of trainable elementary neural networks on the same input, in the role of GPs, and a linear function of their outputs as their combiner. Indeed, our combination stage is rather straightforward, since it consists of a linear regression that minimizes the sum of the square shifts between real and simulated outputs. It is important however to underline that our regression can "see the entire error landscape", because it is trained in a single shot over all the errors of the parcels. This way, regression coefficients are computed taking into account all errors at the same time, rather than epoch-by-epoch as it would happen should the parcels and the combination stage be co-trained. Framing our contrivance in the Social Intelligence sphere, we give some answers to the basic question: to what extent an aggregate of weak learners, like our GPs, may recover the activity of a strong learner in supplying acceptable approximation of a target function?

To this aim, the paper is organized as follows. While Section II provides the general framework of our inference, Section III specializes the framework to our learning by gossip scheme with a special emphasis to: A. the sample complexity of the learning algorithm, B. the reliability of its results and C. the optimality of the involved representations.

The answer may become still more complex if, after the training phase, we subject the GPs to an educational path by optimizing a goal involving ancillary attitudes of theirs. We discuss this extension of the model in Section IV, where we relate the prominence with which the single GPs contribute to the formation of the overall ensemble response to a proper reciprocal positioning of them on a plane. While (long term) training recall genetic mutation along millennia on the human chromosomes, mobility is a sort of epigenetic process inducing (short term) adaptation of GPs to their environment, a process that we may liken to the education of pupils in the human society.

We devoted a large room in Section V to numerically check the value of this contrivance in all its variants. As a result, we provide some hints to appreciate the benefits of our ensemble approach, as a function of the degree of *competence* and *education* of the GPs and of empowering expedients that are peculiar of this framework. As a whole, these hints may represent the conclusion of our paper, that in Section VI we coupled with the statement that our technique can be likened to learning a kernel function in a distributed, data-driven way, as opposed to having a central authority selecting it, e. g. by trial-and-error. We expect this "democratic" perspective to learning to be useful for practical applications as well as in connection to other state-of-the-art methods.

II. The framework of possible explanations

Given a distribution law, think of its unknown parameters as random parameters. In place of a specific prior distribution, they inherit probabilities from a standard source of random seeds that are also at the basis of the random variables they are deputed to characterize. Like with a barrel with two interlocking taps, through the former we get, from the source, samples of the random variable for given parameters; while from the latter we get samples of the parameters for given observed variables. We measure what happens with the former and compute what would happen with the latter. This raises a general inference procedure, that we denote as *Algorithmic Inference* [7, 13], whose basic steps are the following:

1. *Sampling mechanism* $\mathscr{M}_X$. It consists of a pair $\langle Z, g_\theta \rangle$, where the *seed* Z is a random variable without unknown parameters, while the *explaining function* g_θ is a function mapping from samples $\{z_1, \ldots, z_m\}$ of Z to samples $\{x_1, \ldots, x_m\}$ of the random variable X we are interested in. This function is indexed in θ, which represents the not yet set parameters of the random variable, i.e either a scalar value or a vector of values we want to investigate. Thanks to the probability integral transformation theorem [14] we have that, by using the uniform variable U in $[0,1]$ as a seed, the explaining function g_θ for

X distributed according to any continuous distribution (with obvious extension to the discrete case) computes the following mapping:

$$x_i = F_X^{-1}(u_i) \tag{1}$$

where F_X^{-1} is the inverse of the X cumulative distribution function (CDF). Not always U is the most appropriate seed. For instance one prefers using the standard Gaussian variable Ψ as a seed of a Gaussian variable X with mean μ and standard deviation σ through the explaining function $x_i = (\mu + \psi_i \sigma)$, as F_X^{-1} is unavailable in its closed form for this X.

All this falls in the common practice of random variable simulation. The benefit of formalizing it in terms of sampling mechanism $\langle Z, g_\theta \rangle$ lies firstly in a clear partition of what is out of our handling and what may be the object of our inference. We can say nothing new about seeds $\{z_1, \ldots, z_m\}$, since they are randomly tossed from a perfectly known distribution; hence they are completely unquestionable as for the single value, and completely known as for their ensemble properties. On the contrary, the explaining function groups into θ the free parameters that we want to infer from the sample. As for a second benefit, they are exactly the seeds that state links between observations and parameters of a given X. We cannot say which value has θ, since we do not know the seeds of the observations. Rather, we may transfer the probability mass of the seeds from the sample to the parameters realizing the sample – our concept of compatibility – as we will see in the next point.

2. *Master equations.* The actual connection between the model and the observed data is tossed in terms of a set of relations between statistics on the data and unknown parameters that come as a corollary of the sampling mechanism. We call these relations master equations. Pivoting around the statistic $s = h(x_1, \ldots, x_m) = h(g_\theta(z_1), \ldots, g_\theta(z_m))$, where s and h are vectors in their general instantiation, the general form of a system of *master equations* is:

$$s = \rho(\theta; z_1, \ldots, z_m). \tag{2}$$

With this relation we may inspect the values of the parameter θ that *could have generated* a sample with the observed statistic s from a particular setting of the seeds $\{z_1, \ldots, z_m\}$. Hence, if we draw seeds according to their known distribution – uniform in our case – we get a sample of parameters in response [7]. In order to ensure this sample clean properties, we involve sufficient statistics w.r.t. the parameters [15] in the master equations. For instance, let consider a negative exponential r.v. X whose CDF and sampling mechanism are defined as:

$$F_X(x) = 1 - \mathrm{e}^{-\lambda x} I_{[i,\infty)}(x); \qquad x = \frac{-\log u}{\lambda} \tag{3}$$

where u is a seed uniformly drawn in $[0,1]$ and $\lambda \in [0,\infty)$ is the unknown parameter to be inferred on the basis of an m-sized sample x. If we identify as a suitable (well-behaving in [16]) statistic the sufficient statistic $s_\Lambda = \sum_{i=1}^m x_i$, a monotonic non increasing relationship reads:

$$\left(\lambda \leq \widetilde{\lambda}\right) \Leftrightarrow \left(s_\lambda \geq s_{\widetilde{\lambda}}\right) \tag{4}$$

where $s_{\widetilde{\lambda}} = \sum_{i=1}^m \widetilde{x}_i$ and $\widetilde{x}_i$ is the value into which u_i would map if we substitute λ with $\widetilde{\lambda}$ in the explaining function. As S_λ follows a Gamma distribution law with shape and scale parameters respectively m and $1/\lambda$ [17], we have that Λ as well follows a Gamma distribution of parameters m and $1/s_\Lambda$, thanks to the commuting roles of variable specification and parameter between s_λ and λ.

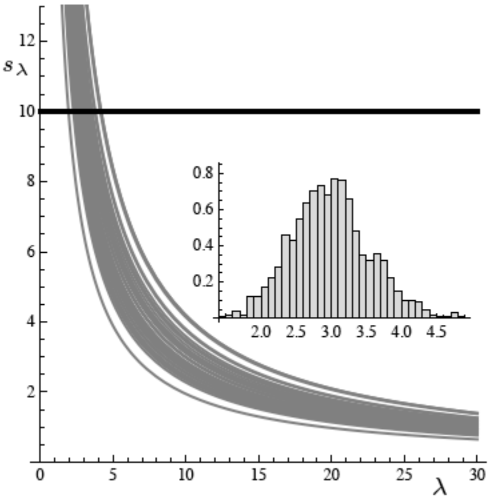


Figure. 1: (a) Course of s_λ w.r.t. λ and histogram of the parameter when $s_\Lambda = 10$

3. *Parameters population.* Having fixed a set of master equations, you may draw seeds in an infinitely large number so as to map from a population of seeds into a population that is representative of the random parameter Θ. The specific features of the mapping translate the uniform distribution of the former into a properly shaped distribution of the latter. In this way we obtain the graph in Figure 1, which reports the distribution law of the random parameter Λ.

A. *The gossip variant*

The goal of gossipers inference is a function f to be regressed from a set of input-output pairs. The regression function is constituted by the questioned GPs ensemble (GPE) where in turn, in the basic version, GPs are *three-layers perceptrons* (feed-forward neural networks (FNN) in general) whose hidden layer is activated by a sigmoidal-logistic function. The output layer is linearly activated and uniformly random bound weights are used in all layers. As shown in Fig. 2 the GP output is sent to an upper node representing the linear combiner producing the GPE output. In this contrivance, seeds are multivariate, as they are represented by the outputs of the GPE. They comply with their definition in Point 1), where a multivariate X is the output of the neural networks that in turn is the seed Z of the variable Y produced by the combiner. Different sampling mechanisms, hence different X distributions, correspond to different weights extractions,

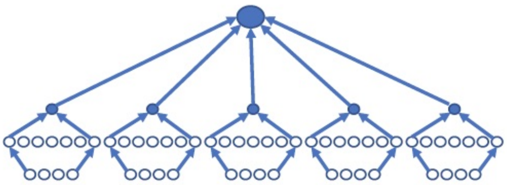


Figure. 2: The basic model

.

and, thank to weights independence, their ensemble is a suitable Z for Y . Things become less straightforward if we decide to train the GPs [18]. Complexity does not derive from a reduction of the parameters' randomness *per se*; actually function g_θ in the sampling mechanism may be completely deterministic too. Rather, it derives from the dependency of the learned parameters on the training set that determines a dependency among seeds as well, as we will discuss in next sections.

III. Learning by gossip

Using the gossip responses Z as seeds has the drawback of working with a unknown seed distribution law that can be recovered only by simulation. On the other hand, computing the output of the questioned function f as a linear function of Z makes the statistic mean square error $S = \sum(t-y)^2$ - with $t =$ the target value and $y =$ the value actually computed by the learned function - to be individually sufficient [17] w.r.t. the related coefficient of the linear function. Rather than on the values of the regression coefficients, we focus directly on S (or analogous ones) to appreciate the quality of our inference. Denoting with Σ the extension of the above sum over the entire Y population, i.e. to the entire input to the learned function, S is an estimate of Σ that we may parametrize as follows

$$S = h(\mathbf{X}, \mathbf{W}) \qquad (5)$$

where $\mathbf{X}$ is a sample in input to f and $\mathbf{W}$ is the set of the three-layer perceptron weights. For fixed $\mathbf{w}$ we face the usual bias-variance trade off. For any training set - test set split of $\mathbf{x}$, different $\mathbf{w}$s denote different representations of the learning problem, among which we may screen the more efficient ones. In accordance to our model, we read the entire GPE contrivance as a sub-symbolic kernelization of a linear regression problem. Kernels are originated by a mapping function ϕ so that the Kernel matrix (for instance, at the bassis of the support vector discrimination) is

$$K_{ij} = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_i)$$

where ϕ is decided *a priori*. In our case, we aim to learn ϕ exactly.

A. Learning as a matter of sample complexity

Coming to a binary version of the problem for the sake of simplicity, let us consider a straight line in the two-dimensional case and a plane in the three-dimensional one, as linear separators dividing positive from negative points (Fig. 3). One of the linear delimiters is the "true" divider — the concept c, the second is our hypothesis h about it. To avoid the growth of the symmetric difference between concept and hypothesis, we need at most 2 or 3 points (we call *sentry points*) that bar a rotation of the hypothesis so as to have another symmetric difference *completely* including the current one.

The key functionality of the sentry points is to bind the expansion of the symmetric difference $c \div h$ through forbidding any rotation of h into a h' pivoted along the intersection of c with h. Whatever the dimensionality n of the embedding space, in principle we would need only 1 point on the border of the angle between c and h, provided we know the target concept c. This point will act as a sentry against this expansion [19]. In fact constraining h' to contain the intersection of h with c gives rise to up to $n-1$ linear relations on h' coefficients, resulting in a single degree of freedom for h' coefficient, i.e. a single sentry point. However, as we do not know c, the chosen sentry point may lie exactly in the intersection between c and h, preventing it to sentinel the expansion of the symmetric difference. So we need one more sentry point, and in general, as many points as the dimensionality of the space. Figures 3(a) and (b) illustrate this concept in case $n = 2$ and $n = 3$ respectively. Sentry points provide a way of characterizing the sample complexity of a class of concepts [20] (hyperplanes, in our case), that is dual to the Vapnick Cervonenkis complexity [21]. There are theorems in the literature that establish the equivalence between these two notions of complexity, but the notion adopted here allows to visualize the role of the points determining the complexity [22]. In our case, the points to be divided are the results of (weakly trained) GPs . This is another way of kernelizing the X space where the original points lie.

In other words, sentry points are minimal sets of points capable of binding the symmetric difference between concepts in a given class and hypotheses generated by a consistent algorithm. Sentry points shown in Fig. 3 are representatives of groups that bind the concept-hypothesis symmetric difference related to the weak learners. Denoting by n_c the number of sentry points of our combiner and by n_{w_i} the one of the i-th GP, the theory says that the number n_t of sentry points of the GPE ensemble is given by [22]

$$n_t \leq n_c \times n_w \qquad (6)$$

Eq. (6) is more binding than the analogous inequality which holds on the growth functions [23]. In our case, the common value n_w of n_{w_i}s is $O(NLog(N))$, where N is the total number of training parameters of the multi-layer perceptron realizing the weakly trained GPs [24]. The detail n_c of a hyperplane (the target of our learning) is equal to the h yperplane dimension. Within this framework, we will now investigate the efficiency of our learning paradigm and its exploitability beyond the usual statistic properties.

B. The reliability of the gossip perspective

From (6) we see that we can modulate our learning effort as a function of the complexity of the target function f. Indeed, the GPE's sample complexity depends on the richness ot this contrivance, hence on the number and structure of its GPs. In turn, the approximation of the trained function g with f depends on the gap between the sample complexity of f and

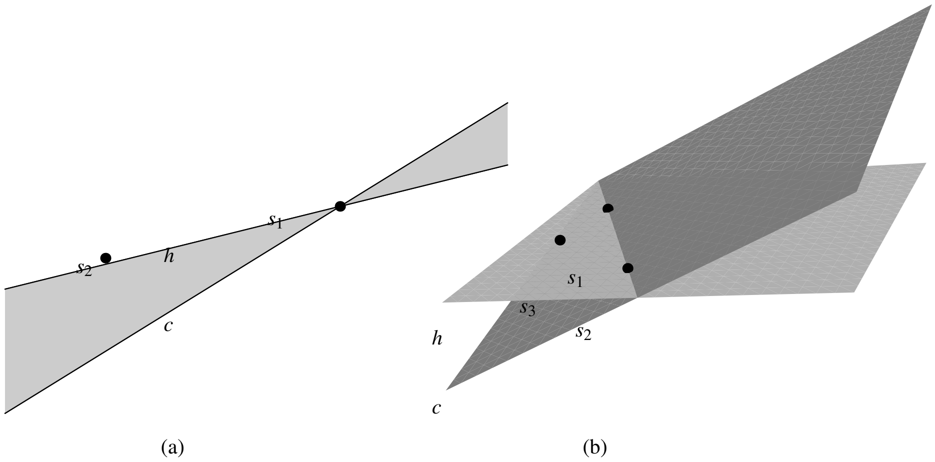


Figure. 3: Sentry points needed in the worst case to bind the symmetric difference between the hypothesis h and the target concept c in (a) two-dimensional and (b) three-dimensional spaces.

the one of g. In synthesis, we may empower the GPE to fill up this gap. This comes, however, at a computational cost determined by the GPE sample complexity and the accuracy with which we decide to train the single GPs, where binding the training time reverts in a smaller n_w. We are basing our computation on a definitely random background: it is constituted by the random initialization of the GP parameters as a reasonable counterpart of the uninformed gossip background in a social network. The latter is randomly biased by the environment of the individual gossiper, its education and surrounding influencers included. Hence our question is: can we rely on the *opinion* of such poor interlocutors? We are assured by both theoretical and experimental arguments. As for the former, our ultimate inference problem is to learn a hyperplane, that is the most elementary function to be inferred. Indeed, VC dimension and detail of a hyperplane are equal to the hyperplane dimensionality d. This makes S extremely close to the Σ with high probability $(1-\eta)$ – that meets our wish, namely:

$$\Sigma \leq S + \sqrt{\frac{d(\log(2N/d)+1-log(\eta/4)}{N}} \tag{7}$$

where N is the sample size. Numerical evidence will be the goal of the next section.

C. Beyond Probability

Probability looks the most convenient tool for describing uncertain situations. Provided you know your sample space and you have enough observations of it, you can organize the observations into statistics so as to reliably infer general properties of the sample space that may prove useful in future occasions. The key is to relate the statistics to the properties in a way that the frequency with which these properties are falsified is asymptotically as small as you want. This is the basis strategy of Algorithmic Inference [7].

Our current framework is notably more complex: we have two families of seeds, one underlying the random input $\mathbf{X}$, another the random weights $\mathbf{W}$ and a sub-symbolic function (the neural network) relating them to the questioned property, i.e. the MSE Σ of the inferred g. This foils any effort to infer the Σ distribution law. Rather, we may try conditioning our operational framework so as to favor low values of Σ.

We acquaint this problem from the *ergodic process* perspective. Let us consider a sequence of random variables $\mathbf{X}$ on the same discrete and finite probability space. It means that the set $\mathfrak{X}$ of values the random variable may assume along the sequence is the same. What changes is the probability distribution on them. Thus, the probability mass of a value is a function of two parameters: the specific value concerned and the step along the sequence at which we are questioning the probability. In the Markov process this step is denoted as a time clock, where the time progress is punctuated by the transform applied to the probability distribution over X. It is a transition matrix M so that $P_{t+1} = MP_t$. In our case $\mathbf{W}$, which is randomly generated at each clock, affects the Σ distribution in a way that is independent of the randomness of $\mathbf{X}$ (since we train on a given $\mathbf{x}$) . In general terms, a process is denoted as ergodic if for any *regular* function ψ the sample mean of ψ along an infinite sampled trajectory of $\mathbf{X}$ equals the expected value of $\psi(\mathbf{X})$ at any clock time. Our goal is to enforce a similar property on the random process of our computation,so that the observation of a sequence of MSE for a fixed $\mathbf{X}$ sample along a long sequence of $\mathbf{W}$ will give us insight about properties of Σ for a fixed $\mathbf{W}$ along a long sequence of $\mathbf{X}$ samples. In this way we aim at identifying an optimal $\mathbf{W}$ which minimizes the generalization MSE, expecting that this optimality will remain in place when further $\mathbf{X}$ samples will be considered. Hence our first question is about the ergodicity of our pseudo-process. From the algorithmic perspective, $\mathbf{X}$ and $\mathbf{W}$ play a symmetric role in the input a to the FNNs hidden layer. For instance, let consider a three (input, hidden, output) layer GP and split $\mathbf{W}$ into $\mathbf{W}_{01}$ weights between input and hidden layer and $\mathbf{W}_{12}$ between hidden and output layer. We simply obtain:

$$\mathbf{a} = \mathbf{W}_{01} \cdot \mathbf{x}$$

Any subsequent computation depends on $\mathbf{a}$, apart from the final linear regression which depends separately on $\mathbf{x}$. Moreover, a proper rescaling of both variables induces similar ranges in the two directions $\mathbf{X}$ and $\mathbf{W}_{01}$, while the prevalence of linear operators leads to normal distributions in both cases. Numerical experiments discussed in the next section confirm this analysis. Hence, given this directions' *twisting* for grant, on a given sample we simulate many GPEs and focus on the one computing the minimum S. We remark that the cumulative distribution of minimal Σ will be biased toward 0 with the number of ensembles — a condition that makes the observed minimum close to the optimum. The relevant question is: ”who tell us that this optimality condition will be preserved on a new instance of $\mathbf{X}$? ”is solved by the closeness of S to Σ highlighted in (7) that allows us to rely on the persistence of optimality. Rather, problems arise from the relations between performances on training and testing. Here the overfitting trap represents again a big issue as it will be shown in the numerical experiments.

IV. Moving gossips

We look for a further improvement of our ensemble solution by weighting the contribution of the GPs. In turn this is achieved by a quick incremental process in a space that is orthogonal to the one of the FNN connection weights that has been exploited during the training phase. Namely, we put the GPs in a plane and endow them with mobility, so that they can pursue a proper positioning w.r.t. the others in order to: 1) get the most favorable position with respect the source of information (the teacher in the training phase) and 2) avoiding neighboring other GPs trained with the same effects (hence assessed with similar parameters — the neural network connection weights — to process the input), thus resulting uselessly redundant [25]. To this aim, the dynamic of the GPs motion is specified as follows.

A. The Lagrangian of GPs

Let us recall the general model [26] referring to an r-layer MLP where all neurons of a layer are located in an Euclidean (two-dimensional, by default) space $\mathcal{X}$. We fix the layout notation, where subscript j refers to neurons lying on layer $\ell+1$ and i,i' to those located in layer ℓ. Namely, on each neuron indexed by i we have:

- an attraction force A by the neurons of the upward layer, which for each j is expressed by:

$$A = G\frac{m_j m_i}{\zeta_{ji}^2} \tag{8}$$

  where G is the gravitational constant and ζ_{ji} is the distance between the two neurons in their role of particles of masses m_i, m_j. The distance is considered in a three-dimensional space, where the third coordinate refers to the distance between layers. We assume it to be a constant h so high that it allows us to resume in it both the contribution of the components in the $\mathcal{X}$ plane and the square root of G, for the sake of simplicity;

- an l-repulsive elastic force R between particles of the same layer which are closer than l, expressed by:

$$R = k_{ii'} \max\{0, l - d_{ii'}\} \tag{9}$$

  where $k_{ii'}$ is the elastic constant between particles i and i'. The force is linearly dependent on the compression $l - d_{ii'}$ between them.

Hence, the cumulated physical energy of the network is the sum of the three terms:

$$L = \xi_{p_1}P_1 + \xi_{p_2}P_2 + \xi_{p_3}P_3 \tag{10}$$

for suitable ξ_{p_i}, with

$$\begin{aligned} P_1 &= \frac{1}{h}\sum_{i,j} m_i m_j && (11)\\ P_2 &= \frac{1}{2}\sum_{i,i'} k_{ii'} \max\{0, l - d_{ii'}\}^2 && (12)\\ P_3 &= \frac{1}{2}\sum_{i} m_i \|v_i\|^2 && (13) \end{aligned}$$

Of the above expressions, the former states the gravitational potential corresponding to (8), the second expresses an l-repulsive elastic energy and the latter the kinetic energy in correspondence to the neuron velocities v_is.

In this *physical* environment the Lagrangian functional ruling the motion of the neuron in the role of particle finds a solution in the Eulerian dynamics. It entails the classical kinematic equations linking the particle position to the accelerations and relates the latter, in turn, to the corresponding conservative force field (see Figure 4). In particular, the acceleration vector for the generic neuron is described as follows:

$$\begin{aligned} a_i = \xi_1 \sum_j m_j \text{sign}(x_j - x_i)) + \\ - \xi_2 \sum_{i'} k_{ii'} \max\{0, l - d_{ii'}\} \text{sign}(x_{i'} - x_i) \end{aligned} \tag{14}$$

for proper ξ_i. Moreover, in order to guide the system toward a stable configuration, we add a viscosity term which is inversely proportional to the actual velocity, namely $-\xi_3 v_i$, which we do not reckon within the Lagrangian addends for the sake of simplicity. In turn, denoting with $x_i^{(n)}, v_i^{(n)}$, and $a_i^{(n)}$ respectively the position, velocity and acceleration of neuron i at instant n (after a suitable time discretization), we come to the usual kinematic equation:

$$x_i^{(n)} = x_i^{(n-1)} + v_i^{(n-1)} t_n + 1/2\, a_i^{(n)} t_n^2 \tag{15}$$

with t_n denoting the length of the n-th time step.

To conclude the model we identify:

- the mass of the neurons with their information content, which in a back-propagation training procedure is represented by the back-piped error term δ. Moreover we get the neuron mass after a suitable normalization in order to maintain constant the total mass on each layer; namely: $m_i = \frac{|\delta_i|}{\|\delta\|_1}$.
- the elastic constant $k_{ii'}$ hinges on how similar the normed weight vectors are, i.e. on the modulus of the cosine of the angle between them:

$$k_{ii'} = \left| \frac{\langle w_i \cdot w_{i'} \rangle}{\|w_i\| \cdot \|w_{i'}\|} \right| \tag{16}$$

We adapt this model to the top layers of our architecture, namely to the layer containing the GP output nodes and the

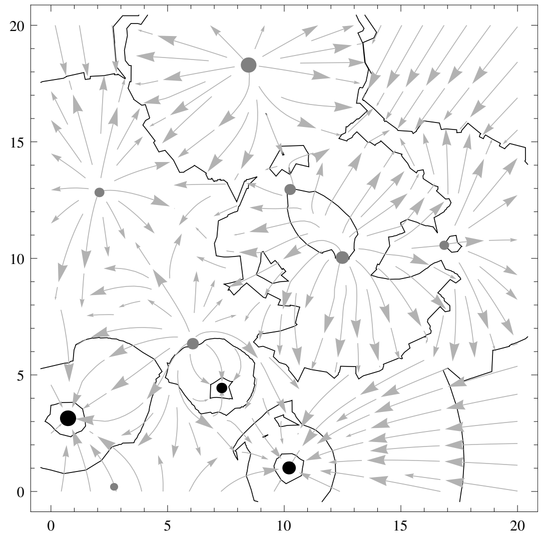


Figure. 4: Potential field generated by both attractive upward neurons (black bullets) and repulsive siblings (gray bullets). The bullet size is proportional to the strength of the field, hence either to the neuron mass (black neurons) or to the assumed outgoing connection weights similarity (gray neurons). Arrows: stream of the potential field; black contour lines: isopotential curves.

uppermost one made up of the node collecting the results of the linear regression on them. Of the above formulas, the one we modified is eq. (14) where we substitute the mass of the upper layer with the one of the lower layer to give sensitivity the dynamics. In turn, the masses of the lower layer simply coincide with the GP output error.

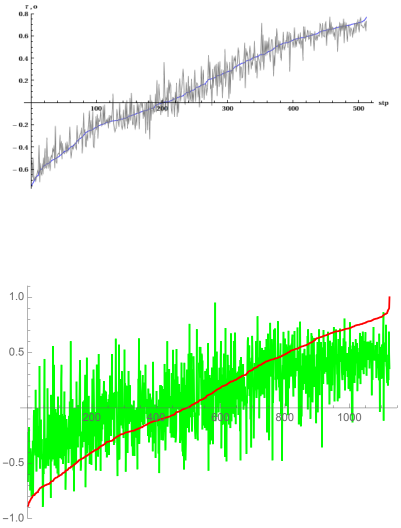


Figure. 5: Errors on Pumadyn regression. Course of the network output with sorted target patterns achieved by (a) a complex neural network, (b) one of the most performing GPE

Minima and extremes				
Experiment	**training**		**testing**	
	min	exteme	min	exteme
$3L\alpha = 0.00005$	0.0738381	0.0755341	0.0818115	0.0885704
$5L\alpha = 0.00005$	0.063183	0.063183	0.0740216	0.0750288
$5L\alpha = 0.0005$	0.063165	0.0634284	0.0656679	0.0694787

Table 1: Features and Conditions

V. Numerical experiments

We based our experiments on the well-knowm Pumadyn benchmark `pumadyn8-nm`. This benchmark is drawn from a family of datasets which are synthetically generated from a Matlab simulation of a robot arm [27]. It contains 4,500 samples, each constituted by 8 inputs and one output. The former record the angular positions and velocities of three arm joints plus the value of two applied torques. The latter is the resulting angular acceleration of one of the joints. This acceleration is a nonlinear function of the inputs which is affected by moderate noise as well. Our reference result is the one obtained some years ago through a special 5 layers FFN where the neurons of a layer are allowed to move inside it to get the most rewarding position with respect to neurons of the upper layer [26]. We appreciate the generalization capability of the network in Fig. 5(a), where we represent in blue the sorted test set targets and in gray the corresponding values computed by our network using both training and testing replicas of size 512, according to DELVE testing scheme [28].

We replace this complex neural network with our contrivance getting results like in Fig. 5(b). Namely, we span a set configurations where we stressed:

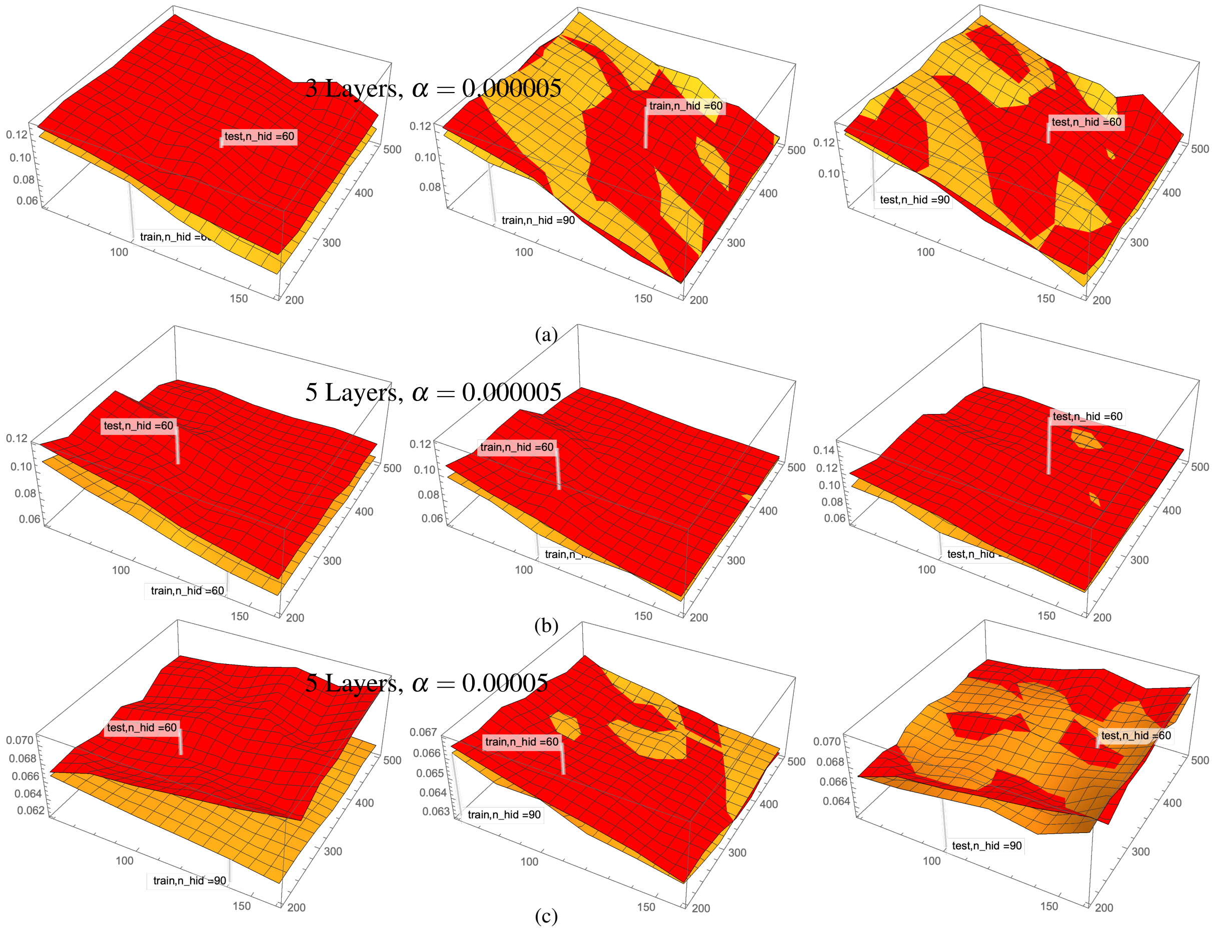
- the GP architecture: either 3Layer $(8, n_h, 1)$ or 5L$(8, n_h, \lfloor n_h/2 \rfloor, \lfloor n_h/3 \rfloor, 1)$
- number n_h of neurons in the main hidden layer, ranging from 60 to 90
- number of GPs, ranging from 60 to 160

We train the GPs in parallel and solve in a single shot the identification of the regression coefficients of the combiner neuron.

We rely on a single train/test split to gain a first sensitivity on the operational parameters that may be synthesized in the surfaces in Fig 6.

The pictures show that enriching the GP architecture generally pays in terms of training and testing error. A smaller learning rate (5×10^{-6}) insures a regular descent of these errors that prevents overfitting. Table 1 denotes that a more aggressive learning (learning rate $= 5 \times 10^{-5}$) may provide somehow smaller errors with a non clear parametrization that produces overfitting, with a divergence betwen the training and testing surfaces. Consequently the minimum does not lie in correspondence of the maximal degrees of freedom and training epochs (where we registered the extreme value). In any case, the more rich GP architecture achieved by the five layer neural network fosters a better performance of the entire GPE.

We also realize that the ensemble significantly improves the skill of the single GP. For instance, the graph in Fig. 7 shows a gain around 1.5, with obviously better benefits with the growth of the training degrees of freedom. Actually this may



3 Layers, $\alpha = 0.000005$

(a)

5 Layers, $\alpha = 0.000005$

(b)

5 Layers, $\alpha = 0.00005$

(c)

Figure. 6: MSE trend with number oh hidden nodes varying fro 60 to 160 and nunber of wphocs vaying fron 200 to 500. Labels on the graph specify the number og GPs, the number of layers and the learning rate.

be considered as the benefit of the consciousness the single GPs gain as a statistical effect of combining at a metalevel the opinions of the other gossipers.

A. Stressing the method

In this paper we are mainly concerned with the reliability of these trends, i.e. on their stability with changing surroundings. For this reason, we:

- Investigated replicas of the experiments as for both the train/test random split and training parameters initialization.
- Run the experiments on two neural network optimizers, respectively supplied by sklearn (*skl*) and TensorFlow (*tf*) packages, with different options on the optimizing algorithm.

While experiments in Fig. 6 have been carried out by using *skl*, we shifted from *skl* to *tf* to carry out more computing-intensive experiments in view of gaining in computational efficiency. Namely, on each grid point we computed 8 replicas of the experiment to appreciate the dispersion of the solutions. On each replica we changed both the random (0.75,0.25) split of the dataset and the random initialization of the GP MLPs. We focused on the $(3L, 5\times10^{-6})$ solution as a much effective and relatively cheap one. The interpolating surface is the 3Dplot joining the mean value of the above points in Fig. 8.
Looking at Fig.8 we can learn some lessons:

1. For the same architecture, investing in computational efforts is generally rewarding, but shows a saturation effect. We may stress all the three parameters, but
   - Over around 300 training epochs no tangible benefits arises
   - The number of hidden layer nodes may be stretched as well but 100 looks a reasonable threshold
   - Number of GPs appear the most sensitive scale parameter tough with analogous saturation.
2. The dispersion of the solutions is limited enough, so justifying the above computational efforts to improve the mean performance
3. A relevant benefit of investing in computational effort is the avoidance of overfitting, that appears more frequent with shrunk architectures (with a few hidden nodes and GPs).

In Fig. 9 we contrasted these results with the analogous ones we got by using *skl*. On the latter, we focused on a single item per each grid node, given the limited dispersion discussed above, and only on the extreme cases got with high numbers of hidden neurons (90). What emerges in Fig.9 is that MSEs obtained with *tf* are a bit better. Both packages implement the same base algorithm, thus differences come from minor strategic choices. For instance, hidden neurons are activateb by a *sigmoidal* function in *tf* and *tanh* in *skl*. Other parameters such batch-size and momentum have been set in a neutral way, i.e. by using exactly the default values of the two packages.
We also remark the different roles of the three operational parameters. Namely, Fig. 10 (a) highlights the preeminent role of n_{GPs} with respect to n_{hidden} and n_{epochs}. In particular, the latter may host overfitting, as shown in fig. 10(b).
Concerning the last point of Section II, we remark that Fig. 11 denotes an asymptotic growth of the correlation between the errors of the various GPs with their training, as a consequence of the uniform efficacy of the training. Whereas the information contribution of the single GPs to the ensemble learning is certified by the well distribution of the spreads around the true values as it emerges from the pictures in Fig. 12.

B. Gossiper promenades

After the training phase, we may enforce further education on each GP by optimizing a goal involving ancillary attitudes. Namely we let the GPs evolve according to the dynamics outlined in Section IV-A. Figure 13 reports the training and test set errors of 6 GPEs with a graduation of their previous training and the companion evolution of their mutual positions. Each GP is constituted by a three layer FFN with 60 hidden neurons. We focused on GPEs made up of 16 members initially located on a square grid. Then, after the training phase, we let them change their positions according to (15) along 100 steps. The five pictures on each line refer to a training phase made up of $tcn = \{20, 70, 120, 170, 220\}$ learning cycles. Given the exiguity of the boiling up parameters the learning rate α has been set to 0.005 Figure 14 differs from Figure 13 only for the position of the attractor, respectively in the center of the grid in the former and under the lower left corner in the latter.
Comparing Figures 13 and 14 with Figures 6 and 8 we may say that education is never a waste, even though it affects only slightly the efficacy of the ensemble. With only 16 GPs and 60 hidden nodes, a proper spatial location of the former reduces the errors both in training and test of around 10%, a relative improvement requiring variously (from 10to 100%) enlarging the hidden nodes in the comparative pictures. A second consideration concerns the attractor position. A position outside the original grid generally foster a better performance than a position inside. Finally, the GP promenades may induce bifurcation, with some trajectory inducing MSE improvements while others may degrade the performance.

C. Looking for representations

Another lesson we may take from our experiments concerns the most effective contributions of the GPs. The idea is that, rather than relying of the output of scarcely skilled GPs we can get better results by looking at what they understand of the input, hence to the hidden neurons outputs that are used to in the regression in place of the final output. This algorithm mode, that we denote as *Share*, produces an enlargement of the regression input, escaping biases contained in the output, and definitely an improvement of the results. We implemented it on the three-layer version of GPs. Figure 15 shows similar figures of Fig. 6, though with a definitely limited number of GPs and hidden neurons. Note that in this case we maintain the learning rate extremely low, $\alpha = 0.000005$

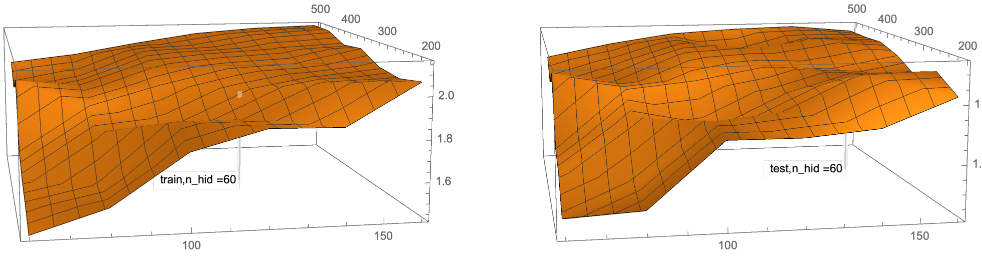


Figure. 7: Ratio of Mean GP training MSE over ensemble training MSE with the number of hidden nodes and of training epochs.

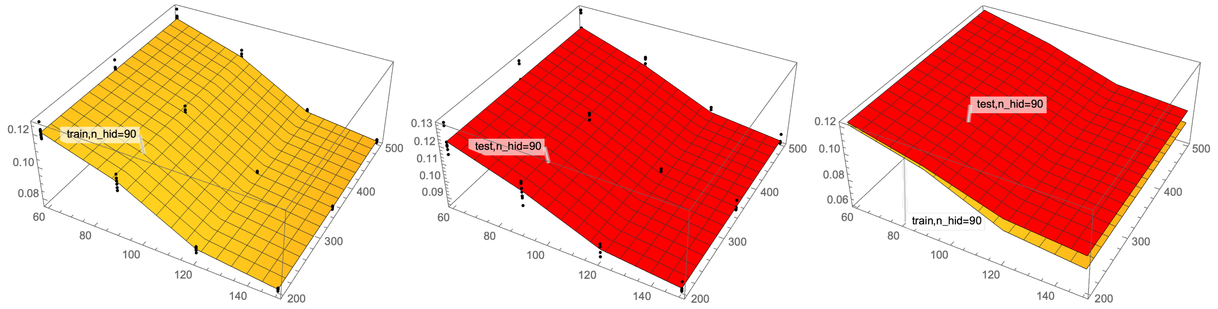


Figure. 8: Same graphs as in Fig. 6(a), but using a different algorithm

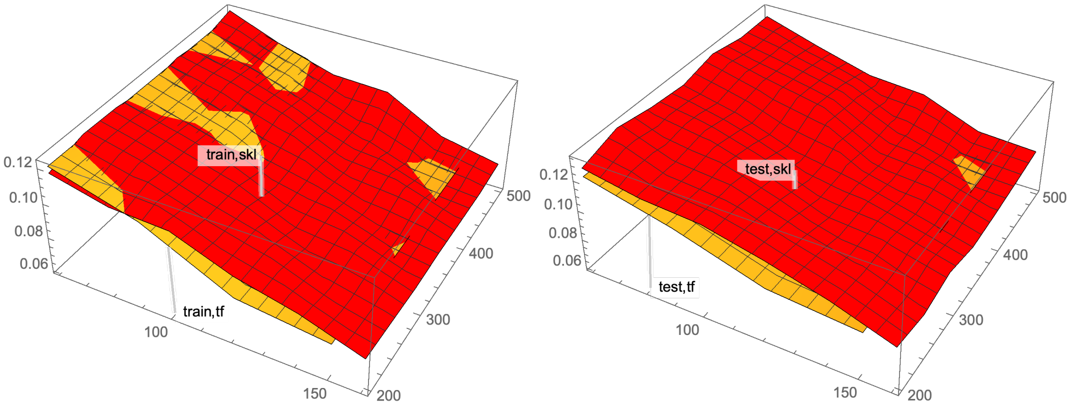


Figure. 9: Contrasting the graphs in Figg. 6(a) and 8

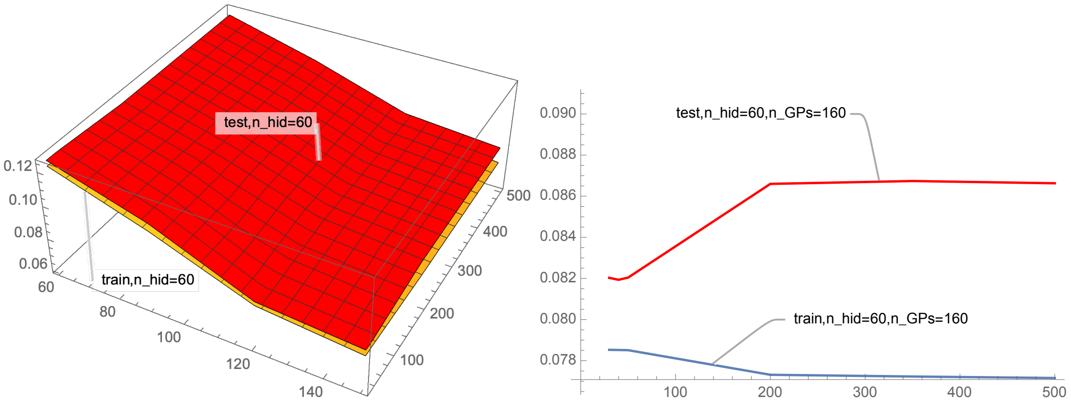


Figure. 10: Test MSE trend with operational parameters in an extended grid ($n_e pochs$ ranging from 30 to 500) in three and two-dimensional spaces.

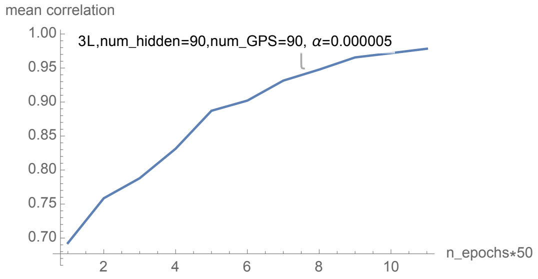


Figure. 11: Growth of the correlation between the errors of the various GPs with their training.

.

because we prefer having a slow evolution of the 60×16 inputs to the linear combiner.

D. Weighting the gossiper's perception

In search of the cumulative effect of representation plus motion, we followed again the trained GPs in their evolution in the positioning space and affect with the out-coming weights exactly the hidden layer outputs generated as in the previous section. Also in this case a proper positioning of the GPs improves the procedure's accuracy, as shown in Figure 16. However it is a matter of fractions of percentage points, as a residual improvement after considering the hidden nodes outputs. Also the trends of the error require consideration, since we have a small degradation of the training error, coupled with a small improvement of the testing error.

E. Exploiting the method

A further lesson concerns *meritocracy*. Does it make sense to focus on the best performing GPEs?
As a preliminary cionsideration, in Fig. 17 we report the empirical CDF of the test set MSE of 60 replicas of the grid point $(n_{hidden} = 90, n_{GPs} = 90, n_{e(pochs)} = 30)$ and of 24 replicas of the grid point $(n_h = 90, n_{GPs} = 90, n_e = 350)$. Both curves denote an almost uniform distribution in a range that depends on the number of training epochs. The different colors refer to different train/test random split of the dataset. Their uniform distributions denote the absence of bias due to the splits.
Fig. 18 fosters a positive answer to our question with some caveats. We consider two instances, the former referring to a GPE made up of 16 GPs, the latter of 60 GPs. In both cases each GP consists of a 5 layers FFN with $n_h = 60$. The data collection strategy was the following. :

1. *Exploratory phase.* Draw an initial sample of size 800 from the training data and of size 400 from the tast data. Generate from the former 100 GPE replicas, They have FFN connection weight and threshold that are initially random (with different seeds) and then trained for n_e epochs with learning rate α. Draw the the training MSE ECDF and the test MSE CDF (red curves in the first column)
2. *Control phase.* From the the training MSE ECDF extract the 50_th quantile and use the corresponding GPE replica as a common replica – the control GPE.
   (a) *Case phase.* Draw 100 further training and test samples (of the above size) and draw the corresponding MSE ECDF (blue curves).
3. From the the training MSE ECDF extract the 20_th quantile and use the corresponding GPE replica as a near optimal replica (too much extreme quantile could refer to anomalous conditions) – the case GPE.
   (a) Draw 100 further training and test samples (of the above size) and draw the corresponding MSE ECDF (green curves).

Since hyperplanes are driven by the sampled values, the second and third graphs refer to populations with greater number of degrees of freedom with respect the former, that generally reflects into a better descent toward minimal MSE – that implies some drift from the ergodicity assumption in section III-C. The first row supports our thesis about *meritocracy*. The graphs in the first columns report the testing MSE in the three phases. Green curve above blue curve denotes that the near optimal ensemble privileges smaller MSE that the common replica, and its median almost coincides with the testing MSE of this replica computed on the initial sample. The second row shows the blue and green curves intertwining. This depends on two factors:

- a longer training of the GPs: 20 epochs in the former experiment versus 200 in the current one. This renders the performances of the case and control ensembles extremely close
- an high value of the learning rate inducing overfitting. Actually we used $\alpha = 0.00005$, which in Fig 6 shown this phenomenon for analogous GPEs.

Column two and three in Fig. 18 highlight this difference. In the first row we have close MSE values in training and testing with a positive correlation denoting coherence. In the second row is the opposite.
Thus, like in the human life, meritocracy needs to be framed in a proper context to avoid unfaithful representations. A key to interpret these situations may be the correlation between training and testing MSE, that in the instance of the first row equals $0,44$, in the second equals -0.61. This value does not disincentivize long trainings, rather asks for careful trainings. For instance the same GPE instance of row two trained with $, \alpha = 10^{-6}$ gives rise to a correlation equal 0.992.

F. Numerical discussion

Having accuracy and computational effort as performance metrics, we may synthesize our numerical analysis through the following considerations:

1. The answer to the basic question: "Can an ensemble of weak learners replace a strong expert?" is: "it depends on the quality of the results we expect", hence on its compatibility with the observed data, contrasted with the effort/competence we want to put on the plate. Coming to our specific case, Fig. 5 clearly establishes that the simplification introduced by the GPE algorithm, and its variants, force us to settle on a relatively coarse

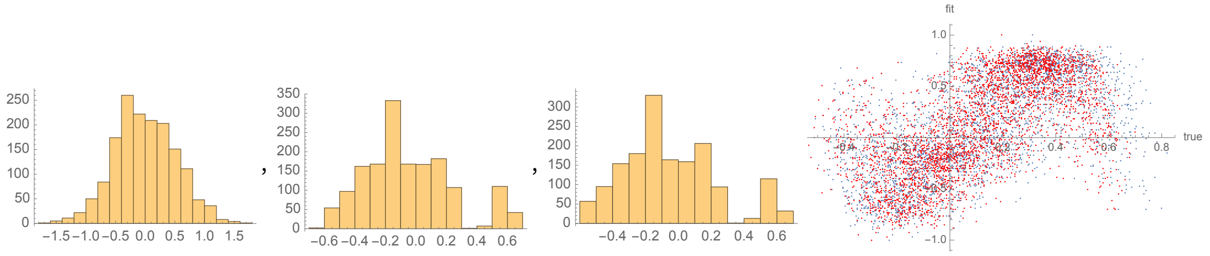


Figure. 12: Hitrogram of the GPS spreads around three true points and true-fitted values scatterplot of two (blue,red) GPs.

.

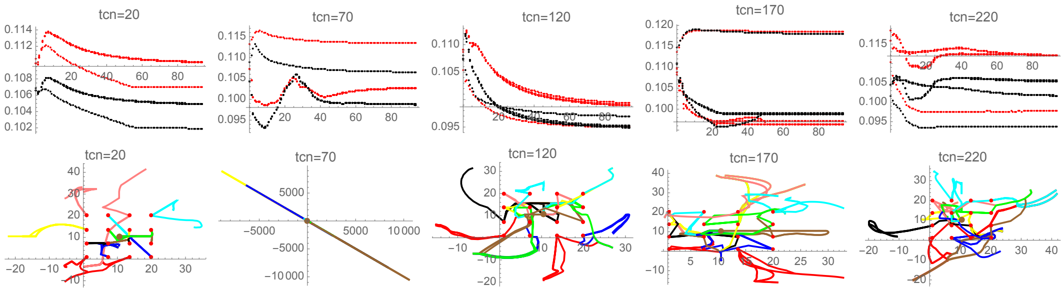


Figure. 13: Trends of moving GPEs. First row: course of MSE with motion steps after the number of training cycles on the headings; black curves: training errors, red curves testing errors. Second row: the companion GP promenades; different colors refers to different GPs; red bullets: initial GP positions, brown bullet: techer position

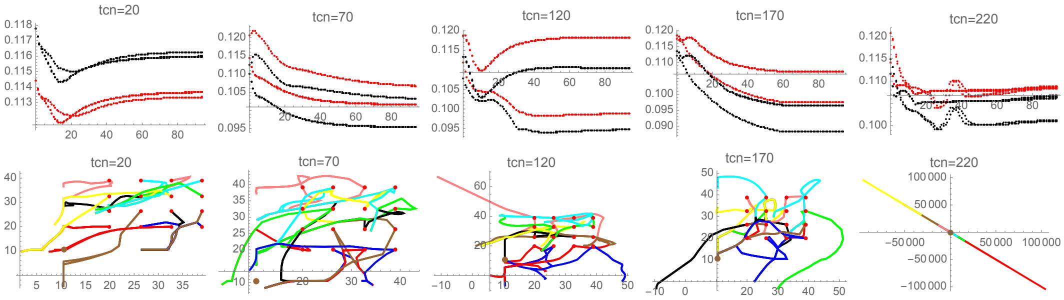


Figure. 14: Same graphs of Fig. 13 but drawn by a different position of the teacher.

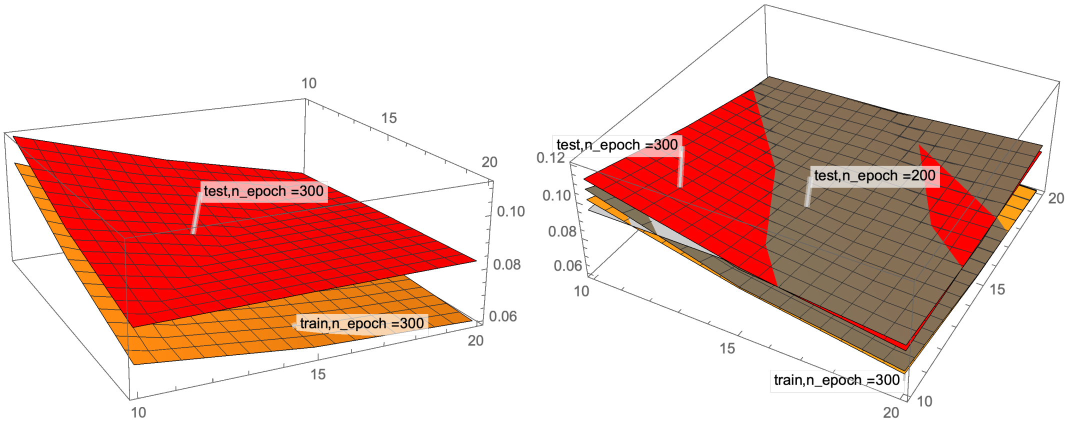


Figure. 15: Similar graphs of Fig. 6 but referred to the Share mode of the algorithm that uses small sewts of GPs and hidden neurons.

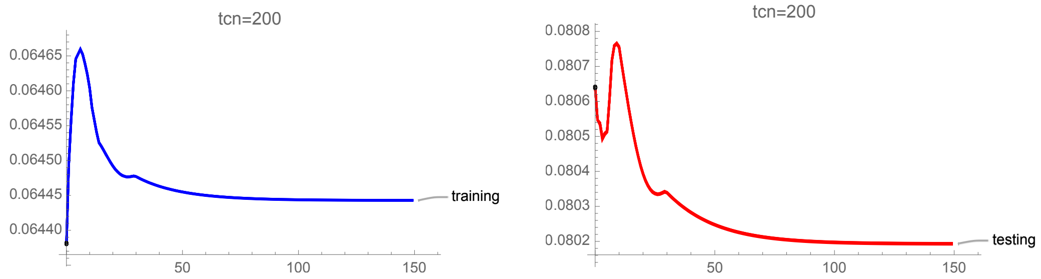


Figure. 16: Similar graphs of Fig. 6 but referred to the Share algorithm with moving GPs and a single setting of the parameters.

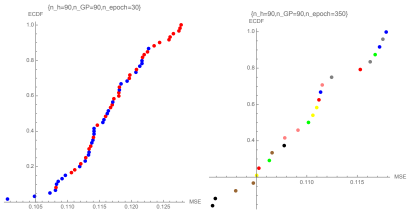


Figure. 17: Empirical cumulative distribution functions of MSE samples on a grid cell. Different colors denote different train/test splits .

approximation of the function compatible with the observed data we were looking for. Actually ranging from the random case (both target and regressed value are independent uniform random variables in $[-1,1]$) till the rather sophisticated way expressed in Fig. 5(b) of regressing the latter on the former, we have an MSE gap with extremes $(0.1667, 0.006)$. The left extreme is not the most unfavorable one, since it does not take into account biases and correlations, the right one is not the most favorable, like we simply may discover in [27]. More closely to our framework and in very rough terms, MSE of a single GP is $O(0.14)$ while MSE of our best contrivance is $O(0.05)$. This denotes an appreciable improvement, yet maintaining our accuracy one order less than the left end of the gap.

2. To bring down the MSE from $O(0.14)$ to $O(0.05)$ we experimented strategies and tools.

   - Regarding strategies, we could establish what follows:

     - when investing on computational resources, privilege member skills (the number of layers of our perceptrons) first and train the members gently (very low learning rate) and appropriately (proper activation functions)
     - for a given skill consider that there is a limit to its improvement coming from both the strength of the architecture (number of hidden nodes) and the length of the training. Above a given strength and length, improvements become intangible. The values of these thresholds increase with the complexity of the architecture.

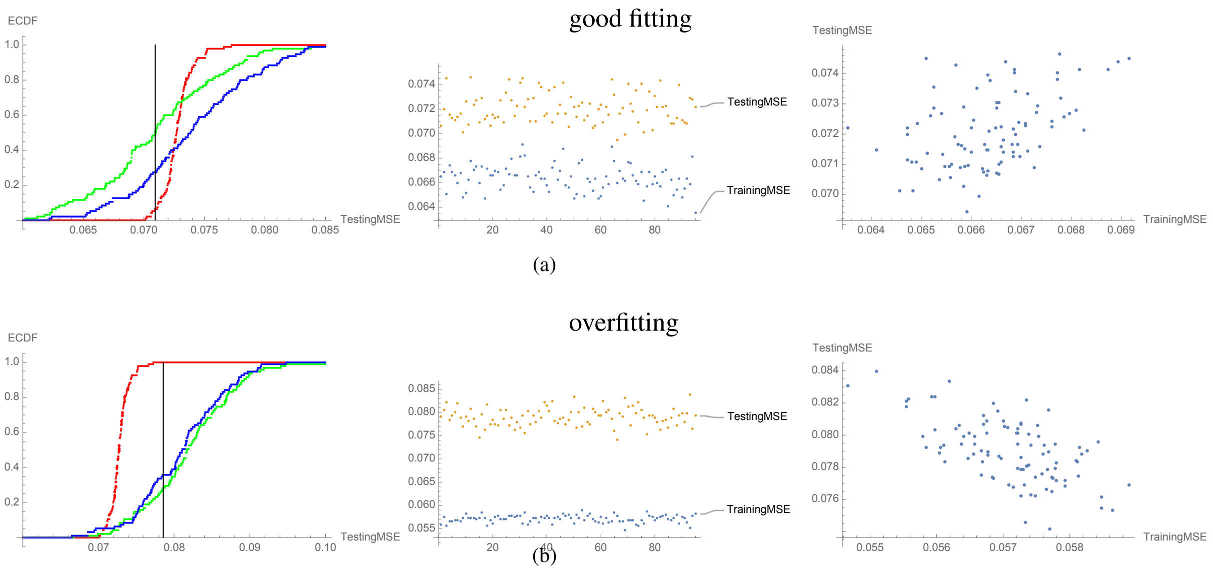
3. Regarding tools, we experimented two variants of the GPE algorithm in terms of:

   (a) Given the weakness of the GPs, as a counterpart of the search for local simplicity and save of resources, we may opt for issuing the final response of the collective brain on the way the GPs *perceive* the external solicitations (the input of the regression problem) rather than on the poor evaluation they can produce of these inputs. Thus we may base the final regression on the output of the hidden neurons of a small number of GPs getting results that are comparable with those obtained with one order large number of GPs.

   (b) To reduce the weakness of the GPs, we educate them, after training, just by inducing them to search the best spatial position with respect to the teacher (the proper pupil and chair desks in a classroom). This is done in order to attribute a proper weight to the GPs output to produce the ensemble result. This may produce a further 10% decrease of MSE.

VI. Conclusions

Trying to understand which kind of Collective Brain we are de facto implementing today, and, above all, the benefits/drawbacks coming from this ecosystem, we stressed the CB paradigm on a controllable environment where we can quantitatively appreciate efforts and performances. Numerical experiments on the well know benchmark represented by Pumadyn regression allows us establishing clear indications and caveats on the collective brain management.
Besides delving into quantitative aspects, we highlighted a methodological one as well, concerning learning of subsymbolic kernels. Kernelized spaces are a formidable resource to get rid of some learning problems. However the identification of a proper kernel remains in the domain of intuition and trial-and-error methods, with the further problem of rendering kernel implementation as less costly as possible [29]. Our approach provides a way of *learning* the mapping functions originating the kernels with a computational effort that remains under our control. This is not a minor task, since a proper kernelization leads us beyond probabilistic evaluations. We used a simple numerical experiment to show that ”a kernel is for life” provided some ergodic conditions are satisfied, so that if we experiment a good kernel in the training phase, we are almost sure of its perfomance also in the test phase



(a)

(b)

Figure. 18: Two instances in search of optimal representations.

What we propose in this paper is just the start of a research line that we plan to develop in the future with more extensive both theoretical and numerical investigations. We concentrate on a single benchmark to appreciate many nuances of the approach. We expect interesting results by implementing the approach in both sociological frameworks in the field of social networks and in approximate computation tasks to allow a variety of computational resources to be accessible to a vast community of AI workers.

References

[1] I. McGilchrist, “Reciprocal organization of the cerebral hemispheres.” *Dialogues Clin Neurosci.*, vol. 12, no. 4, pp. 503–515, 2010.

[2] S. Nelson, P.and Zyglidopoulos, “Learning from foundation: Asimov’s psychohistory and the limits of organization theory.” *Organization*, no. 4, pp. 591–608, 1999.

[3] A. Dhanhani, E. Damiani, R. Mizouni, and D. Wang, “Analysis of shapelet transform usage in traffic event detection,” in *IEEE Cognitive Computing Congress*, pp. 41–48.

[4] B. Apolloni and F. Kurfess, Eds., *From Synapses to Rules – Discovering Symbolic Rules from Neural Processed Data*. New York: Kluwer Academic/Plenum Publishers, 2002.

[5] T. G. Dietterich, “Ensemble methods in machine learning,” in *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15.

[6] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1249

[7] A. B., M. D, and G. S, *Algorithmic Inference in Machine Learning*. ADELAIDE – AUS: Advanced Knowledge International Pty, 2006, vol. 5.

[8] A. B., D. Malchiodi, and J. Taylor, “Learning by gossip: a principled information exchange model in social networks,” *COGNITIVE COMPUTATION*, vol. 5, pp. 327–339, 2013. [Online]. Available: http://dx.medra.org/10.1007/s12559-013-9211-6

[9] B. Apolloni, C. Aitis, and M. Maffetti, “Watching by gossip,” *Internet of Things*, vol. 3-4, pp. 90 – 103, 2018.

[10] G.-B. Huang and L. Chen, “Convex incremental extreme learning machine,” *Neurocomputing*, vol. 70, no. 16, pp. 3056 – 3062, 2007.

[11] R. M. Parkavi, M. Shanthi, and M. C. Bhuvaneshwari, “Recent Trends in ELM and MLELM : A review,” *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 1, pp. 69–75, 2017. [Online]. Available: http://astesj.com/archive/volume-2/volume-2-issue-1/recent-trends-elm-mlelm-review/

[12] M. Lukoševičius and H. Jaeger, “Survey: Reservoir computing approaches to recurrent neural network training,” *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, Aug. 2009.

[13] B. Apolloni and S. Bassis, “The randomness of the inferred parameters. a machine learning framework for computing confidence regions.” *Information Sciences*, vol. 453, pp. 239 – 262, 2018.

[14] M. Rosenblatt, “Remarks on a multivariate transformation,” *The Annals of Mathematical Statistics*, vol. 3, no. 23, pp. 470–472, 1952.

[15] S. Stigler, “Studies in the history of probability and statistics. laplace, fisher and the discovery of the concept of sufficiency”,,” *Biometrika*, vol. 3, no. 60, pp. 439–445, 1973.

[16] A. B., W. Pedrycz, S. Bassis, and D. Malchiodi, *The Puzzle of Granular Computing*. Berlin: SPRINGER, 2008. [Online]. Available: http://dx.medra.org/10.1007/978-3-540-79864-4

[17] S. S. Wilks, *Mathematical Statistics*, ser. Wiley Publications in Statistics. New York: John Wiley & Sons, 1962.

[18] B. Apolloni, A. Al Shehhi, and E. Damiani, “Bargaining compatible explanations,” in *2019 IEEE International Conference on Cognitive Computing (ICCC)*, July 2019, pp. 98–105.

[19] B. Apolloni and S. Chiaravalli, “Pac learning of concept classes through the boundaries of their items,” *Theoretical Computer Science*, vol. 172, pp. 91–120, 1997.

[20] B. Apolloni, S. Bassis, and D. Malchiodi, “Compatible worlds,” *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 12, pp. e2883 – e2901, 2009.

[21] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[22] B. Apolloni and D. Malchiodi, “Gaining degrees of freedom in subsymbolic learning,” *Theoretical Computer Science*, vol. 255, pp. 295–321, 2001.

[23] Y. S. Abu-Mostafa, “Hints and the vc dimension,” *Neural Computation*, vol. 5, no. 2, pp. 278–288, 1993.

[24] E. B. Baum and D. Haussler, “What size net gives valid generalization?” *Neural Computation*, vol. 1, no. 1, pp. 151–160, 1989.

[25] B. Apolloni, “A possible hamiltonian of social communities dynamics,” *J Comput Eng Inf Technol*, vol. 5, no. 2.

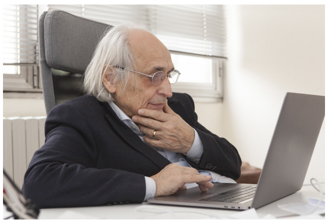
[26] B. Apolloni, S. Bassis, and L. Valerio, “Training a network of mobile neurons,” *The 2011 International Joint Conference on Neural Networks*, pp. 1683–1691, 2011.

[27] P. I. Corke, “A robotics toolbox for matlab,” *IEEE Robotics Automation Magazine*, vol. 3, no. 1, pp. 24–32, 1996.

[28] C. Rasmussen, R. Neal, G. Hinton, D. van Camp, Z. Revow, M. andGhaharamani, and R. Kustra, Z. R.and Tibshirani, “The delve manual,” Department of Computer Science, University of Toronto, Canada, 1996. [Online]. Available: http://www.cs.toronto.edu/ delve/

[29] N. Cesa-Bianchi, Y. Mansour, and O. Shamir, “On the complexity of learning with kernels,” *CoRR*, vol. abs/1411.1158, 2014. [Online]. Available: http://arxiv.org/abs/1411.1158

Author Biographies



Bruno Apolloni was born in Naples, Italy, on October 16th, 1946. He's a retired full professor on Computer Science. He taught last 30 years at the University of Milano, Italy. His research interests are in the frontier between probability, mathematical statistics and computer science, with special regard to statistical bases of learning, neural networks, granular computing, and dynamical processes in biology. He introduced the Algorithmic Inference approach in statistics as a conceptual and methodological tool to solve modern computational learning problems with the massive use of computers. He also introduced some non-Markov processes to model intentionality in various-scale biological systems, from bacteria colonies to social networks. He published more than one hundred scientific papers.

Ernesto Damiani is the Senior Director of Artificial Intelligence and Intelligent Systems Institute, Khalifa University, leader of the Big Data area at Etisalat British Telecom Innovation Center, and Full Professor at Università degli Studi di Milano, where he leads the SESAR Lab. Ernesto Damiani's work has more than 15,500 citations on Google Scholar and more than 6,100 citations on Scopus, with an h-index of 34. His areas of interest include Artificial Intelligence, Machine Learning, Big Data Analytics, Edge/Cloud security and performance, and cyber-physical systems. Ernesto has been a recipient of the Stephen Yau Award from the Service Society, of the Outstanding contributions Award from IFIP TC2, of the Chester-Sall Award from IEEE IES, and of a doctorate honoris causa from INSA – Lyon (France) for his contribution to Big Data teaching and research.