Expert Systems With Applications 71 (2017) 206–215

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Increase attractor capacity using an ensembled neural network

Mario González[a,b,*], David Dominguez[c], Ángel Sánchez[d], Francisco B. Rodríguez[c]

[a] FICA, Universidad de las Américas, Quito, Ecuador
[b] FACI, Universidad Estatal de Milagro, Ecuador
[c] Grupo de Neurocomputación Biológica, Dpto. de Ingeniería Informática, Escuela Politécnica Superior, Universidad Autónoma de Madrid, 28049 Madrid, Spain
[d] ETSII, Universidad Rey Juan Carlos, 28933 Madrid, Spain

ARTICLE INFO

Article history:
Received 14 July 2016
Revised 22 November 2016
Accepted 24 November 2016
Available online 25 November 2016

Keywords:
Hopfield-type network
Synaptic dilution
Network wiring cost
Storage capacity
Ensemble of Attractor Neural Networks
Divide-and-conquer parallelizing

ABSTRACT

This work presents an ensemble of Attractor Neural Networks (ANN) modules, that increases the patterns' storage, at similar computational cost when compared with a single-module ANN system. We build the ensemble of ANN components, and divide the uniform random patterns' set into disjoint subsets during the learning stage, such that each subset is assigned to a different component. In this way, a larger overall number of patterns can be stored by the ANN ensemble, where each of its modules has a moderate pattern load, being able to retrieve its corresponding assigned subset with the desired quality. Allowing some noise in the retrieval, we are able to recall a larger number of patterns while discriminating between pattern subsets assigned to each component in the ensemble. We showed that the ANN ensemble system with $N = 10^4$ units is able to approximately triple the maximal capacity of the single ANN, with similar wiring costs. We tested the modularized ANN ensemble for different levels of component dilution, by keeping constant the wiring costs. This approach could be implemented, for instance, with parallel computing in order to tackle computational costly real-world problems.

1. Introduction

Attractor Neural Networks (ANN) have been used as models for many cognitive functions such as memory (Wills, Lever, Cacucci, Burgess, & OKeefe, 2005), visual field data classification (Fink, 2004), motor behavior (Stringer, Rolls, Trappenberg, & de Araujo, 2003), and for recall and recognition (Ruppin & Yeshurun, 1991). Also, in engineering problems for shape recognition (Amit & Mascaro, 1999), automotive traffic video analysis (González, Dominguez, & Sánchez, 2011), and fingerprint retrieval (Gonzalez, Dominguez, Rodriguez, & Sanchez, 2014) to mention a few. An important research topic is the study of mechanisms underlying memory and the estimation of the information capacity in biological systems (Johansson, Rehn, & Lansner, 2006), as well as, for bio-inspired engineering applications. For any practical purpose a high load of stored patterns at high quality of retrieval is desired, while keeping the computational cost low.

The original Hopfield model (Hopfield, 1982), is a fully connected single layer associative network, with each neuron connected to every other neuron. The fully connected ANN requires complete and symmetric connectivity, and self-interactions are not permitted. Full connectivity is expensive in terms of wiring, as well as in terms of computational costs, given that it requires extensive internal feedback between node units. Diluted networks have demonstrated to increase the pattern load per connection, while using less computational resources. That is, the retrieval capacity in terms of the pattern load $\alpha = P/K$ increases at lower wiring and computational cost, as have been shown for diluted ANN with finite connectivity, for both symmetric and nonsymmetric cases (Arenzon & Lemke, 1994; Brunel, 1993; Derrida, Gardner, & Zippelius, 1987; Sompolinsky, 1986; Wemmenhove & Coolen, 2003). Also, the biological plausibility of diluted connectivity have been deeply studied (Brunel, 2003; Rolls & Webb, 2012; Roudi & Treves, 2008; Wang, Tegner, & Constantinidis, 2004). In this sense, to overcome both wiring and computational costs, diluted networks on different connectivity topologies have been extensively studied (Dominguez et al., 2012; Dominguez, González, Serrano, & Rodríguez, 2009; Dominguez, Koroutchev, Serrano, & Rodríguez, 2007; González, Dominguez, & Rodríguez, 2009).

Therefore if one has a diluted ANN which is in the limit storage capacity, one can exploit the retrieval increase in terms of α, using an ensemble of ANN components with higher dilution. Consequently, each of these components can process a different pat-

* Corresponding author.
E-mail addresses: mario.gonzalez.rodriguez@udla.edu.ec, marsgr6@gmail.com (M. González), david.dominguez@uam.es (D. Dominguez), angel.sanchez@urjc.es (Á. Sánchez), f.rodriguez@uam.es (F.B. Rodríguez).

http://dx.doi.org/10.1016/j.eswa.2016.11.035
0957-4174/

terns subset in order to increase the overall number of patterns that can be stored when compared with the single ANN system at its critical load capacity. It is worth to note that the ANN ensemble's wiring computational cost is the same when compared with the corresponding single-module ANN system.

Another useful characteristic of using a diluted modularized ANN ensemble, is that the transition from perfect retrieval ($m = 1$) to non-retrieval ($m = 0$) is continuous for each module, as opposed to the network discontinuous transition for the fully connected case. Each module in the ensemble goes smoothly from near perfect retrieval ($m = 1$) for low pattern load, to non retrieval ($m = 0$) for high pattern load. For intermediate values of pattern load, one has high retrieval with some $m < 1$ (Dominguez et al., 2007). Taking advantage of this property, one can build an ensemble of ANN modules that can store a higher pattern load with $m < 1$, at similar computational and wiring costs as the single moderately diluted network.

The insight behind the ensemble of ANN components is the divide-and-conquer principle, where a large problem (i.e. the pattern set) is divided into smaller pieces (i.e. pattern subsets). The subsets are learned separately, then, during the retrieval process the sub-networks are combined into a full retrieval of the original set of patterns. Similar approaches have been used for constrained optimization problems (Noel & Szu, 1997). Unlike works dealing with ensemble of feed-forward neural networks where base inducers are constructed using different sampling techniques, and using winner-takes-all or voting techniques (Lopez, Melin, & Castillo, 2007), in this work, one uses a pure divide-and-conquer approach. That is, each subnetwork specializes in learning a disjoint pattern subset, and by adding all the ANN components the retrieval capacity of the ensemble is increased. Note that, the new ANN ensemble maintains the connectivity cost of the "big" ANN system and even when the system capacity of n components is a sum of the disconnected n components, the combined storage capacity of these n components is larger than the capacity of a single densely connected network (see Section 2.4).

Recent works highlight how the modularity and hierarchical structures in associative networks allow to replicate parallel pattern retrieval and multitasking abilities exhibited by complex neural and immune systems (Agliari, Barra, Galluzzi, Guerra, & Moauro, 2012; Agliari et al., 2015a; 2015b; 2015c; Sollich, Tantari, Annibale, & Barra, 2014). These findings could lead to significant development for both theoretical and practical applications. In this way, the motivation of this work is to increase the information processing (storage and retrieval) capacities using an ensemble of diluted ANN modular components. The hypothesis is that an ensemble of n ANN components (each with low connectivity degree K_b), is able to store as much or more patterns, as "big" networks (with high degree $K = n \times K_b$) while taking similar wiring/computational resources. In this sense the present approach could be very useful to be implemented with parallel computing in order to tackle computational costly problems (Bhagat & Deodhare, 2006). It is worth to mention that when dividing problems into subproblems, one may overcome the sensitivity of the model to cross-correlation among the memorized patterns. Even for the uniform random patterns used in this paper, the correlation is not exactly zero due to fluctuations in the generation process (Amit, 1989; Hertz, Krogh, & Palmer, 1991). However, when using an ANN ensemble one can keep the number of stored patterns per module low (compared with the size of the network), thus, the noise becomes negligible by keeping the memorized patterns as stable fixed points. This would allow the ANN ensemble to deal with real-world data where information is structured and correlated such as fingerprint recognition problems (Gonzalez et al., 2014). Also, through the divide-and-conquer parallelizing approach one can apply the modularized ANN to computational costly problems such as video traffic analysis (González et al., 2011).

The rest of the paper is organized as follows. The model is rigorously described in Section 2 by detailing the pattern uniform random coding, the single network topology and dynamics, as well as the proposed ANN ensemble system, the information measures, and computational cost. Then, in Section 3 simulation results are presented and discussed by comparing a single ANN to the equivalent ANN ensemble, for different network parameters, as dilution and number of components in the ANN ensemble. Finally, Section 4 concludes the paper summarizing and discussing the main results, as well as giving future lines of work.

2. The model

In this section the proposed ANN ensemble model is rigorously described, starting with the neural coding, the network topology, and both learning and retrieval dynamics. A schematic representation of the modularized ANN system is illustrated. Then, the information measures of the network and ensemble system are defined. Finally, the computational complexity of our proposal is explained.

2.1. Coding, topology and dynamics

At any given time t, the network state is defined by a set of binary neurons $\vec{\sigma}^t = \{\sigma_i^t \in \{-1, +1\}, i = 1, \ldots, N\}$. The purpose of the network is to recover a set of independent patterns $\{\vec{\xi}^\mu,\ \mu = 1, \ldots, P\}$ that have been stored by a learning process. Each pattern, $\vec{\xi}^\mu = \{\xi_i^\mu \in \{-1, +1\}, i = 1, \ldots, N\}$, is a set of site-independent unbiased binary random variables, $p(\xi_i^\mu = \{-1, +1\}) = 1/2$.

The synaptic couplings between the neurons i and j are given by the adjacency matrix $J_{ij} \equiv C_{ij}W_{ij}$, where the topology matrix $\mathbf{C} = \{C_{ij}\}$ describes the connection structure of the neural network and in $\mathbf{W} = \{W_{ij}\}$ are the learning weights. The topology matrix is built by random links connecting each neuron to K others uniformly distributed in the network (Erdös & Rényi, 1959). The network topology is then characterized by the *connectivity* ratio defined by $\gamma = K/N$. An extremely diluted network is obtained as $\gamma \to 0$, and the storage cost of this network is $\|J\| = N \times K$ if the matrix $\mathbf{J}$ is implemented as an adjacency list of K neighbors. The matrix $\mathbf{J}$ is considered to be symmetrical, i.e. $J_{ij} = J_{ji}$.

The task of the network is to retrieve a pattern, $\vec{\xi} \equiv \vec{\xi^\mu}$, starting from a neuron state $\vec{\sigma}^0$ which is close to it. This is achieved through the neuron dynamics

$$\sigma_i^{t+1} = \text{sign}(h_i^t), \tag{1}$$

$$h_i^t \equiv \frac{1}{K}\sum_j J_{ij}\sigma_j^t,\ i = 1, \ldots, N, \tag{2}$$

where h_i^t denotes the local field of neuron i at time t.

The learning algorithm updates the weight matrix $\mathbf{W}$ according to the Hebb's rule,

$$W_{ij}^\mu = W_{ij}^{\mu-1} + \xi_i^\mu \xi_j^\mu. \tag{3}$$

Weights start at $W_{ij}^0 = 0$ and after P learning steps, they reach the value $W_{ij} = \sum_\mu^P \xi_i^\mu \xi_j^\mu$. The learning stage displays slow dynamics, being stationary within the time scale of the faster retrieval stage in Eq. (1).

2.2. Retrieval overlap measures

In order to evaluate the network retrieval performance, two measures are considered: the global overlap and the load ratio. The overlap is used as a measure of information, which is adequate to

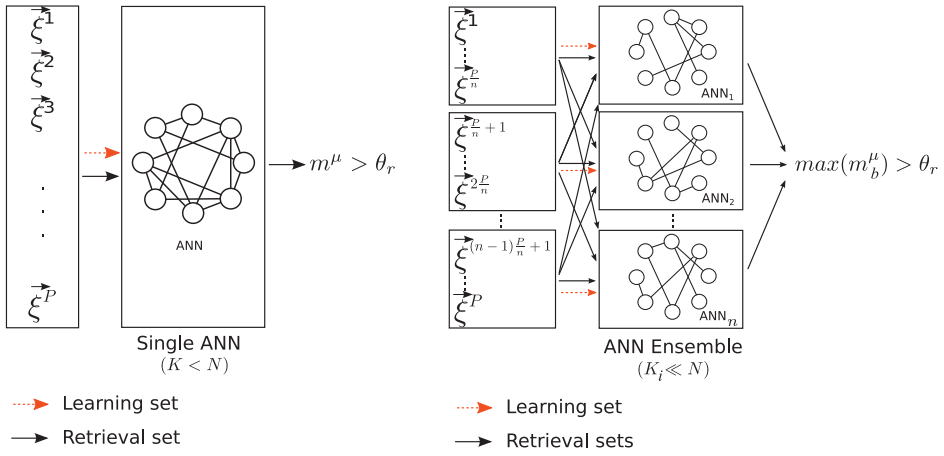

Fig. 1. Schematic representation of a single ANN (left) vs. an ANN ensemble (right) with n components. Red dashed lines represent the learning patterns set flow, black solid lines represent the retrieval patterns sets flow. Overall connectivity in both cases is the same with $K = n \times K_b$, $K_b \ll N$. Retrieval threshold θ_r is the same for both cases. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

describe instantaneously the network's ability to retrieve each pattern. In this case, the overlap m^μ between the stationary neural state $\vec{\sigma^*}$ and the corresponding pattern $\vec{\xi^\mu}$ is:

$$m^\mu \equiv \frac{1}{N}\sum_i^N \xi_i^\mu \sigma_i^*, \tag{4}$$

which is the normalized statistical correlation between the learned pattern $\vec{\xi^\mu}$ and the stationary neural state $\vec{\sigma^*}$ after a sufficient enough long time.

One lets the network evolve according to Eqs. (1) and (2), and measures the overlap between the network states and the patterns. When the overlap between a given pattern and the corresponding neural states of the network is $m = 1$, the network has retrieved the pattern without noise. When the global overlap $m = 0$, the network carries no macroscopic order. In this case, the corresponding pattern cannot be retrieved. For intermediate values of m, where $0 < m < 1$, the pattern has been recovered with a given level of noise $(1 - m)$.

One is also interested in the load ratio $\alpha \equiv P/K$, that accounts for the storage capacity of the network. When the number of stored patterns increases, the noise due to interference between patterns also increases and the network is not able to retrieve them. Thus, the overlap m goes to zero. A good trade-off between a negligible noise (i.e. $1 - m \sim 0$) and the storage of a large pattern set (i.e. a high value of α) is desirable for any practical purpose model.

2.3. ANN ensemble

A schematic representation of the single ANN is presented in Fig. 1-left. The connectivity ratio γ is diluted with $K < N$. A set of P patterns $\boldsymbol{\xi}$ is presented to the network in a learning phase, represented with the red dashed arrow. Then, this set of patterns is presented in a retrieval phase in order to test the recall abilities of the network in terms of the retrieved patterns load α, and the quality of the retrieval m. This is represented with the solid black arrow.

In Fig. 1-right, a schematic representation of an ensemble of ANN modules with a number of n components is presented. The connectivity in each ANN_b module b is highly diluted with $K_b \ll N$, $b \in \{1, \ldots, n\}$. Note that, in order to keep the computational cost of the single ANN and the ANN ensemble the same, one uses $K = K_b \times n$. The set of patterns is divided into disjoints subsets of uniform size $P_b = P/n$, and each pattern subset is learned by its corresponding ANN_b module as represented with the red dashed arrows. For example, $\{\vec{\xi^\mu},\ \mu = 1, \ldots, P/n\}$ for the first module ANN_b, $b = 1$, as shown in Fig. 1-right. The solid black arrows in Fig. 1-right, represent the retrieval stage, in which all the pattern subsets are presented to all ANN modules in order to test the discrimination among them. The target patterns are considered as retrieved by the ANN module with the higher overlap value over the retrieval threshold θ_r, i.e. $\max(m_b^\mu) > \theta_r$. For comparison purpose, θ_r is assumed to take the same value for each component in the ensemble, as well as, for the single ANN system.

2.4. Retrieval measures of the ANN ensemble

In order to evaluate the ensemble performance, one may define the retrieval efficiency R, as the number of learned patterns that are successfully retrieved $R = \frac{P_r}{P_l}$, where P_r is the overall number of retrieved patterns that satisfy $m^\mu > \theta_r$, and P_l is the overall number of patterns presented to the network during the learning phase. One has that $P_l \geq P_r$. When the super-index b is used, P_r^b, P_l^b refer to the ANN_b module b in the ensemble. Here θ_r is the retrieval threshold. The mean retrieval overlap M is calculated as the mean retrieval overlap over all patterns subset $\mu \in 1, 2, \ldots, P_l$, $M = \langle m^\mu \rangle = 1/P_l \sum_{\mu=1}^{P_l} m^\mu$. It is worth noting that in the case of the ANN ensemble, the retrieval pattern load is calculated as $\alpha_R = \frac{P_r}{K_b \times n}$, where n is the number of subnetworks. Thus, we use $K_b \times n = K$ constant for all network ensembles studied, where K is the connectivity of the single "dense" network. Also, it is of worth to define the pattern gain G of the ANN ensemble by taking the single ANN system retrieval performance in terms of recovered patterns (P_r^s) as baseline, and it is given by $G = P_r^e / P_r^s$. Here P_r^e stands for the number of total recovered patterns by the ANN ensemble and P_r^s stands for the patterns recovered by the single network at the maximum retrieval pattern load $max(\alpha_R)$ (see Fig. 4).

For a highly diluted connectivity the pattern storage is moderate. Although, if one combines several ANN modules in an ensemble learning process, one can increase the overall number of retrieved patterns. The statement of this work is that one can increase the stored number of patterns α, with a good quality of retrieval m using an ensemble of ANN modules, with similar computational costs, when compared with a single less-diluted ANN.

In order to measure the performance for comparison purpose, both the ANN ensemble and the single ANN, should have comparable computational costs. This is detailed in the following subsection.

2.5. ANN ensemble computational complexity

The computational complexity in terms of learning is of the order $O(N \times K \times P)$ for the single ANN, where N is the number of nodes, K the network degree, and P the learned patterns. For the ANN ensemble of n components, the computational complexity is n times the single ANN module but with $K_b = K/n, P_b = P/n$. Each component b in the ensemble has a complexity of $O(N \times K_b \times P_b)$, then, for the whole ANN ensemble system with n components, one has a computational complexity of $O(N \times (K/n) \times P)$. Note that the number of neurons N is fixed for the ensemble, and the neuron states are assigned dynamically for the patterns being processed. The computational complexity of each ANN module of the ensemble during learning stage can be expressed algorithmically, as shown in the following pseudo-code.

ANN ensemble computational complexity (learning stage)

The computational complexity for a single ANN module during the learning stage can be expressed as follows. P: number of patterns, N: number of neurons, K: network degree. $PS(P,N)$: pattern set (size $P\times N$). $S(N)$: network state imprinted by pattern μ during learning. W(N,K): weights matrix (started as zero matrix). $C(N,K)$: connectivity matrix.

For $\mu = 1$ to P
$S = PS(\mu)$
For $ni = 1$ to N
For $ki = 1$ to K
$W(ni, ki) = W(ni, ki) + S(ni) \times S(C(ni, ki))$

The computational complexity of the ANN ensemble system is equal to the computational complexity of a single "big" ANN divided by a factor of n, where n is the number of components in the ensemble.

It is worth to note that, the memory cost is the same for both the single ANN system and the ANN ensemble, given that $W = N \times K$ for the single ANN has identical wiring cost compared with the ANN ensemble cost $W = N \times K_b \times n$, $K_b = K/n$ with n components. Please note that the computational simulations use the standard adjacency list representation for graphs. In this representation each vertex is associated in the graph with the collection of its neighboring (adjacent) vertices, i.e. edges. Therefore using this structure, the sum of wiring costs of all n-components in the ANN ensemble is equal to the single ANN in terms of number of connections K.

ANN ensemble computational complexity (retrieval stage)

The computational complexity at the retrieval stage can be expressed in the following pseudo-code for a single component in the ensemble. P:=patterns, N:=neurons, K:=degree, T:=time steps. $PS(P,N)$:=patterns set. $S(N)$:=network state at time step t_i. $W(N,K)$:=weights matrix, $C(N,K)$:=connectivity matrix. nF:=local field of neuron ni (is calculated dynamically).

For $\mu = 1$ to P
$S = PS(\mu)$
For $ti = 1$ to T
For $ni = 1$ to N
$nF = 0$
For $ki = 1$ to K
$nF = nF + W(ni, ki) \times S(C(ni, ki))$
$S(ni) = \text{sign}(nF)$

The retrieval computational cost is similar for both the single ANN and the ensemble with n-components, since the runtime steps for the recurrent update of the network are the same for both cases. The reader must be aware that during the process of patterns identification one has to check all learned patterns P in each component of ensemble resulting in $P \times n$ tests. Nevertheless, one must take into account that the retrieval process of patterns in each diluted module carries less runtime. This runtime reduction is due to the fact that the calculation of the local field of a given neuron has only $K_b = K/n$ terms in the summation (see Eqs. (1) and (2)), compared with the K terms for the single ANN.

3. Simulation results

The present section details the results attained using the proposed ANN ensemble system. A comparison between the single ANN and the ANN ensemble performance is presented for the retrieval of uniform random pattern sets. Then, the ensemble system performance is explored for different parameters, such as the number of ensemble components n and the dilution of the modules K. The retrieval phase of the ensemble system is summarized for the aforementioned parameters in terms of the information measures defined in Section 2. The retrieval performance is analyzed in terms of the mutual information of the ensemble retrieval process. Finally, an analysis of the ensemble performance is presented in terms of the retrieval threshold parameter θ_r.

3.1. Single ANN vs. ensembled ANN retrieval performance

Fig. 2 -left shows the retrieval performance of a single ANN with $N = 10^4$ for different dilution conditions. From $K = 50$ to $K = 6400$, for a retrieval performance ranging from a critical load of $\alpha_c \sim 15/50 \sim 0.3$ to $\alpha_c \sim 992/6400 \sim 0.155$ respectively, for $m^\mu > 0.9$. One can appreciate that the more diluted the network is, the better the performance is in terms of the pattern load α. Given that the value $\alpha = P/K$ depends on the network degree K, since $P = \alpha \times K$, for larger values of K, more patterns (larger P) can be retrieved, thus the curves become denser. However, this occurs at the expense of increasing the computational cost of the single network.

As reproduced by Dominguez et al. (2007), for fully connected networks $\gamma = 1.0$, the transition between retrieval and non retrieval phases is discontinuous going abruptly from $m \sim 1.0$ to $m \sim 0.0$. For the fully connected network, the transition occurs at the critical load of $\alpha_c \sim 0.14$ (Amit, Gutfreund, & Sompolinsky, 1987). Using extremely diluted random connectivity $\gamma \to 0.0$, one should expect a critical load as large as $\alpha_c \sim 0.6366$ for the nonsymmetric case (Derrida et al., 1987). For intermediate dilutions, as shown in Fig. 2-left, one has $\alpha_c \sim 0.155$ for $K = 6400, \gamma = 0.64$, and $\alpha_c \sim 0.5$ for $K = 50, \gamma \sim 0.008$ (the dotted lines are a visual guide). However for the latter case, one has to make the trade-off of considering some noise in the retrieval at values of $m^\mu \sim 0.6$.

One can consider a good enough value of m^μ in order to consider the pattern as retrieved, as long as the value m^μ is significantly larger for the intended pattern μ than for the rest of patterns in the set. Thus, one could set a retrieval threshold $m^\mu > \theta_r$, above which retrieval is considered. This is shown in Fig. 2-right, considering $\theta_r = 0.5$, $R = 96\%$ of the patterns are retrieved above this threshold. The value of $\theta_r = 0.5$ will be used in the rest of the paper for the single ANN and the ANN ensemble. In Fig. 2-right, the retrieval for the ANN ensemble of $n = 128$ components, each with degree $K_b = 50$ and pattern load $\alpha_b = 0.46$ is shown. In the figure are depicted the first and final three components, but the behavior is similar for all 128 networks. One can appreciate that each component of the ANN ensemble retrieves its corresponding subset of patterns successfully while discriminating from other subsets. Taking advantage of the continuous transition one can build an ensemble of ANN modules, with the purpose to retrieve a larger number of patterns, with a retrieval overlap $m < 1$. In this way, keeping the computational cost constant with $K = n \times K_b$, one can increase the number of stored patterns, when compared with a single ANN system.

Fig. 3 -top panels (A and B) show a single ANN system with $N = 10^4, K = 6400, n = 1$ compared with an ANN ensemble system of n components with $N = 10^4, K_b = 100, n = 64$ and $N = 10^4, K_b =$

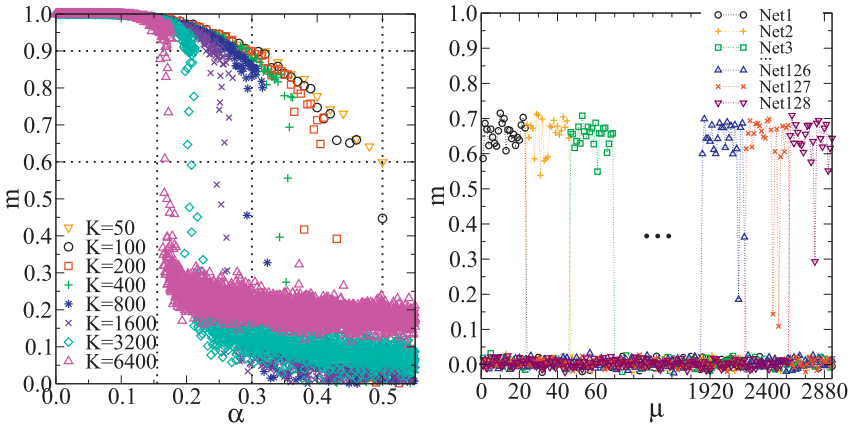

Fig. 2. Left: Single ANN retrieval performance as function of number of neighbors K. Performance decays with increasing K, from $\alpha_c \sim 0.3$ for $K = 50$ to $\alpha_c \sim 0.155$ for $K = 6400$ with $m^\mu > \theta_r, \theta_r = 0.9$. Right: Performance for an ANN ensemble with $n = 128$ networks with $K_b = 50$ neighbors. Pattern load $\alpha_b \sim 0.46$ for each network in the ensemble. Recovered patterns $P = \alpha_b \times K_b \times n = 1280$ with mean retrieval overlap $M \sim 0.6$ with $m^\mu > \theta_r, \theta_r = 0.5$. $N = 10^4$ in both panels. *Note:* μ stands for the index of the learned pattern, the dotted lines are visual guides. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

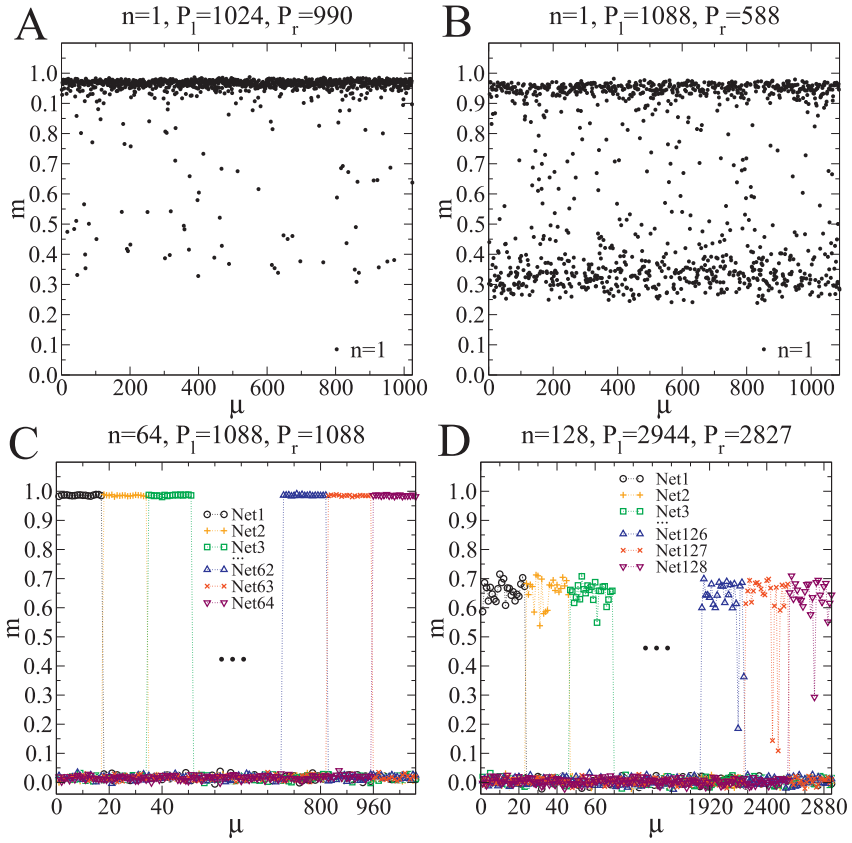

Fig. 3. Top panels: Network retrieval for $n = 1, K_b = 6400$. Retrieval efficiency $R = 990/1024 = 0.96$ (panel A), $R = 588/1088 = 0.54$ (panel B). Bottom panels: Performance for an ANN ensemble with $n = 64, K_b = 100$, $R = 1.0$ with mean overlap $M \sim 0.99$ (panel C). Ensemble with $n = 128, K_b = 50$ (panel D), $R = 0.96$ with $M \sim 0.64$. See Table 1 and description on text for a review of the parameters. $N = 10^4$ and $m^\mu > \theta_r, \theta_r = 0.5$ in all panels, μ stands for the index of the learned pattern. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$50, n = 128$ in the respective bottom panels (C and D). See Table 1 for a detailed comparison of the parameters used in each panel. Table 1 shows the parameter values of the simulation depicted in Fig. 3 panels labeled as A, B, C, and D. One can appreciate for each ensemble, the number of components n, the degree K_b, the pattern load $\alpha_b = P_l^b/K_b$, the learned patterns P_l^b, the retrieved patters P_r^b for $m^\mu > \theta_r, \theta_r = 0.5$, respectively. It is clearly appreciated that the wiring cost $K = n \times K_b$ is constant for all cases, $K = 6400$. One can partition the single ANN system into an ensemble of higher diluted components of equivalent complexity in terms of connec-

Table 1
Parameter review of each panel in Fig. 3. n: number of components, $K_b = K/N$: component connectivity where K is the single ANN system connectivity, P_l^b: learned patterns by component, α_b: load by component, P_r^b: retrieved pattern by component, α_R: retrieval load by system (single ANN or ANN ensemble), P_r: recovered patterns by system, R: retrieval efficiency by system.

Panel	n	K_b	P_l^b	α_b	P_r^b	α_R	P_r	R%
A	1	6400	1024	0.16	990	0.155	990	97
B	1	6400	1088	0.17	588	0.092	588	54
C	64	100	17	0.17	17	0.170	1088	100
D	128	50	23	0.46	22	0.442	2827	96

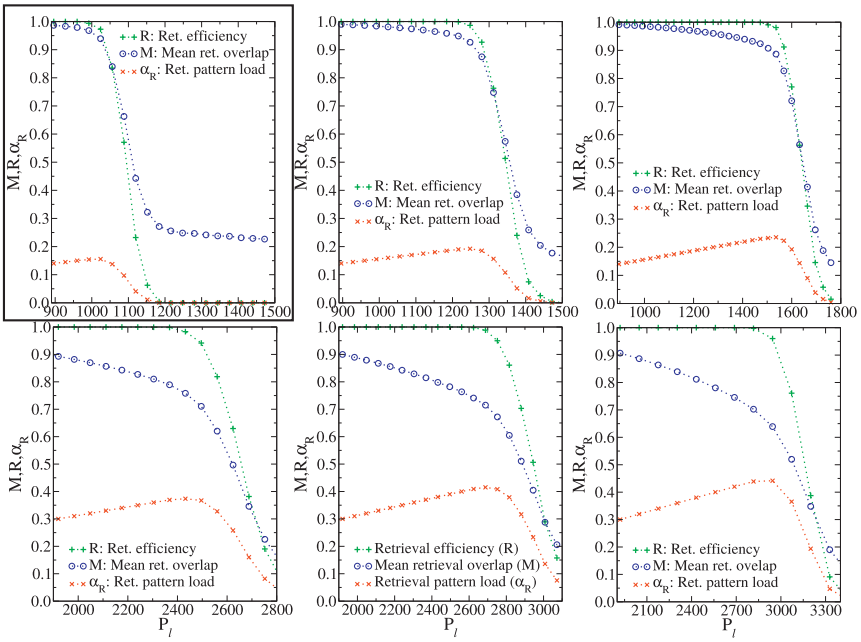

Fig. 4. Retrieval efficiency (R), mean retrieval overlap (M), and retrieval pattern load (α_R), for a network/ensemble with $N = 10^4$ and constant cost $K = n \times K_b$. Recovered patterns considered for $m^\mu > 0.5$. Top panels. Left: Single ANN with $K = 6400$, framed as reference. Middle: ANN Ensemble with $K_b = 3200, n = 2$. Right: ANN Ensemble with $K_b = 1600, n = 4$. Bottom panels. Left: ANN Ensemble with $K_b = 200, n = 32$. Center: ANN Ensemble with $K_b = 100, n = 64$. Right: ANN Ensemble with $K_b = 50, n = 128$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

tivity cost, such that the ensemble system is better in terms of storage capacity, given that each independent component can store more than the P/N patterns from the single ANN system. Although losing quality in the retrieval, the ensemble clearly discriminates the recovered patterns for each component.

In panel A, the pattern load of the network $\alpha = 0.16$ is slightly above its saturation point $\alpha_c \sim 0.155$, which is equivalent to learn and retrieve $P \sim 992$ patterns. The network is provided with $P_l = 1024$ patterns and manages to retrieve $P_r = 990$ of them, that is an efficiency of $R = P_r/P_l = 96.68\%$. Further increasing the number of patterns to learn to $P_l = 1088$, the single ANN system performance declines retrieving only $R = 54\%$ of the learned patterns. This can be appreciated as the cloud of dots with $m^\mu < \theta_r = 0.5$, depicted in Fig. 3 panel B (top right). As commented before, the larger the connectivity degree of the network, the more discontinuous is the transition, as is the case for panels A and B. An increment in the number of patterns from $P_l = 1024$ to $P_l = 1088$ greatly worsens the performance of the "big" network (single ANN system).

On the other hand, in the case of the ANN ensemble, Fig. 3 panel C (bottom left) depicts an ANN ensemble with similar computational cost, with $N = 10^4, K_b = 100, n = 64$, and pattern load $P_l^b = 17$ for each component, that is $\alpha_b = 0.17$, recovering each $P_r^b = 17$ giving a total recovered pattern count (P_r) by the ANN ensemble of $P_r = P_r^b * n = 1088$ with efficiency $R = 100\%$, and mean retrieval quality $M \sim 0.99$. That is, each component of the ensemble is able to retrieve its corresponding pattern subset with a high quality of retrieval. The ANN ensemble system retrieves the totality of the patterns with high retrieval overlap quality at the same computational costs. In panel D (at the bottom right) is depicted the ANN ensemble increasing the learning load to $P_l^b = 23$ patterns that is $\alpha_b = 0.46$, and keeping the computational costs equal with $N = 10^4, K_b = 50, n = 128$. The network achieves the retrieval of $R \sim 96\%$ of the patterns retrieving $P_r = P_r^b \times n = 2827$ of $P_l = 2944$ possible patterns, with a mean retrieval overlap of $M \sim 0.64$. Although the retrieval quality decreased, the network was able of discriminating among each pattern set assigned to each component in the ensemble, as can be appreciated in Fig. 3 bottom right. This was possible because of the continuous transition in α for very diluted connectivity as is the case for each component in the ensemble, allowing discrimination and retrieval with values of $m_b < 1$.

3.2. ANN ensemble retrieval performance vs. number of components

In the top-left panel of Fig. 4 is depicted the single ANN system with $K = 6400, n = 1$, and framed as reference for comparison with an ensemble of $K_b = 3200, n = 2$ and $K_b = 1600, n = 4$ networks, middle and right panels, respectively. The curves for the retrieval efficiency (R), mean retrieval overlap (M) and retrieval pattern load (α_R) are depicted in each panel. A pattern is considered as retrieved for $m^\mu > 0.5$. However, the maximum value of the pattern load α_R curve is considered as the saturation point of the system (being it, the single ANN or the ANN ensemble). At this maximum value α_R(max), the retrieval efficiency is compared for each system. One can appreciate that for the single ANN with $K = 6400, n = 1$ the number of retrieved patterns is $P_r = 1024$, for the $K_b = 3200, n = 2$ and $K_b = 1600, n = 4$ ensemble the number of patterns recovered at the maximum α_R is $P_r = 1231$ and $P_r = 1507$ respectively. For the latter case this implies a pattern gain of $G = P_r^e/P_r^s = 1507/990 \sim 1.52$, that is, the ensemble of $n = 4$ networks/components is able to recover 52% more patterns than the single ANN system.

In the bottom panels of Fig. 4, a similar analysis is performed for ANN ensembles with a larger number of components $K_b = 200, n = 32$, $K_b = 100, n = 64$ and $K_b = 50, n = 128$ in the left, middle and right panels respectively. The pattern gain achieved with the $K_b = 50, n = 128$ ensemble is $G = P_r^e/P_r^s = 2827/990 \sim 2.86$, almost tripling the number of patterns that the single "big" ANN system can store. This level of retrieval is reached with a mean retrieval overlap $M \sim 0.63$ for the $K_b = 50, n = 128$ case, with a retrieval efficiency of $R \sim 96\%$. Note that, the patterns in each component of the ensemble can be clearly discriminated from the other subsets as shown in Fig. 3 bottom panels.

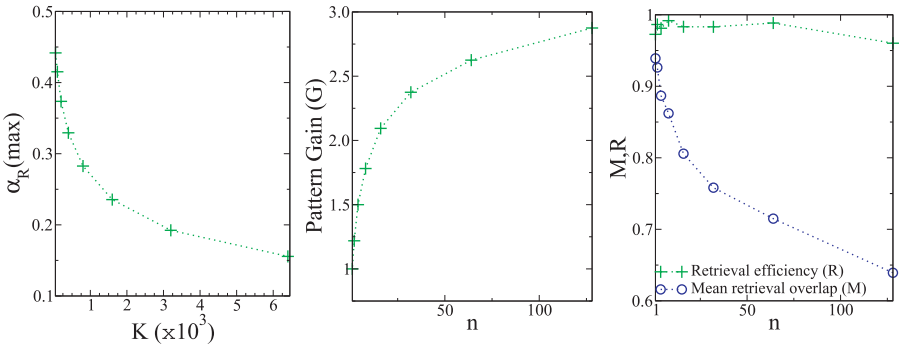

Fig. 5. Left: Retrieval pattern load α_R for different values of connectivity degree K_b of the components in the ANN ensemble. Middle: Pattern gain G for different values of n components in the ensemble. Right: Retrieval efficiency (R) and mean retrieval overlap (M) for different values of components n in the ensemble.

3.3. ANN ensemble retrieval phase diagrams

Fig. 5 summarizes the results for the different performance measures of the ANN ensemble system. Fig. 5-left panel shows the behavior of the systems under study depicting the $\alpha_R(\max)$ for different levels of connectivity degree K_b of each component in the ensemble. One can appreciate that the maximum retrieval pattern load $\alpha_R(\max)$ that the network ensemble is able to achieve is better for extremely diluted components (with low K_b) when compared with not so diluted components (larger K_b). In Fig. 5-middle panel is presented the ensemble pattern gain G for different number of components n. The $K_b = 400, n = 16$ ensemble achieve a pattern gain of $G = 2108/990 = 2.13$, effectively doubling the patterns recovered by the ensemble when compared with a single network of similar computational cost. The $K_b = 50, n = 128$ ensemble achieves a pattern gain of $G = 2827/990 = 2.86$, also with similar costs. Again, the ANN ensemble is able to increase the number of patterns that can be processed. In Fig. 5-right panel the ensemble's retrieval efficiency R and mean retrieval overlap M are depicted for different number of components n. The mean retrieval M overlap decreases for a large number of n, that is for very diluted components. Although, the patterns are recovered with $m^\mu < 1$, the patterns can be effectively distinguished by the component to which they have been assigned, and the retrieval efficiency remains high for the modularized ANN ensemble.

3.4. ANN ensemble information capacity

One can describe the information capacity of the ANN ensemble calculating the mutual information MI of the ensembled retrieval process between the stored patterns and the neural states given by the mean retrieval overlap M. An approximation from Dominguez and Bollé (1998), Dominguez et al. (2009, 2007) can be used to calculate the mutual information MI in terms of this mean retrieval overlap M. This will give

$$MI[M] = 1 - S[M],$$
$$S[M] = -\frac{1+M}{2}\log_2\frac{1+M}{2} - \frac{1-M}{2}\log_2\frac{1-M}{2}. \quad (5)$$

Hence the information ratio can be defined in terms of α_R and M as

$$i_M(\alpha_R, M) \equiv \alpha_R MI[M]. \quad (6)$$

In Fig. 6-left panel is depicted the information ratio defined in Eq. (6) for the different values of components n in the ensemble. One can appreciate that the information ratio i_M increases with the number of components, that is the information ratio is minimum for $n = 1$ and maximum for $n = 128$, clearly increasing with n. This behavior is summarized in Fig. 6-middle panel, depicting the maximum i_M for each curve in Fig. 6-left panel and its corresponding α_R. Both α_R and i_M increase with the number of components n, as shown in Fig. 6-middle panel. In Fig. 6-right panel is depicted the Mutual Information MI calculated in Eq. (5). One can appreciate a trade-off where despite more patterns are allowed to be stored, the information MI decreases when increasing the number of components n in the ensemble. This is because of less information per pattern is stored in the ensemble for very diluted components (e.g. $n = 128, K = 50$), with a lower value of the mean retrieval overlap M. Again as a remark, when considering the number of patterns that the ensemble can store α_R, one has a greater information ratio i_M, for larger number of components n in the ensemble, as depicted in the first two panels of Fig. 6.

Until now we have considered the retrieval threshold, $m^\mu > \theta_r = 0.5$. In next section we will investigate the role of parameter θ_r in the retrieval performance of the network ensemble.

3.5. ANN ensemble retrieval performance vs. retrieval threshold (θ_r)

Fig. 7 depicts the analysis of the ANN ensemble retrieval performance in terms of the value of the retrieval threshold θ_r, for the different number of components n studied in this work. The reader should remind that $\alpha_R(\max)$ is selected, in Fig. 4, in order to get the values of M, R, and the pattern gain G for the each n-components ensemble depicted in Fig. 5. Please note that Figs. 4 and 5 are calculated considering that the network recovers a pattern when $m^\mu > \theta_r = 0.5$. Now we examine different values of parameter θ_r in order to consider a pattern as retrieved by the network. Therefore in Fig. 7 for values of $\theta_r = \{0.5, 0.6, 0.7, 0.8, 0.9\}$, and the maximal retrieval pattern load $\alpha_R(\max)$ (top-left panel) the pattern gain G (top-right panel) the mean retrieval overlap M and the retrieval efficiency R are shown vs. the number of components n in the ensemble. One can appreciate that increasing the number of components n leads to larger values of the maximum pattern retrieval $\alpha_R(max)$, as well as to an increased pattern gain G. This behavior is true for all values of θ_r, but one yields higher $\alpha_R(\max)$ and gain G for lower values of θ_r. This is at the cost of losing quality of retrieval as is shown in Fig. 7 bottom-left panel. The mean pattern retrieval overlap M is depicted for different values of θ_r. One can appreciate that the larger the number of components in the ensemble the lower the mean quality of retrieval M. The trade-off between the storage capacity and retrieval quality by means of θ_r is apparent. Using the maximal value of α_R to calculate the ANN ensemble performance measures M, R and gain G, one gets a fair value for the ensemble retrieval abilities. As shown in Fig. 7 bottom-right panel, a high efficiency of retrieved pattern R is assured for all values of θ_r. This high retrieval efficiency is at the cost of reducing the gain G (top-right) in the ensemble and the quality of retrieval M (bottom-left). However, for all θ_r values, increasing the components n in the ANN ensemble leads to bet-

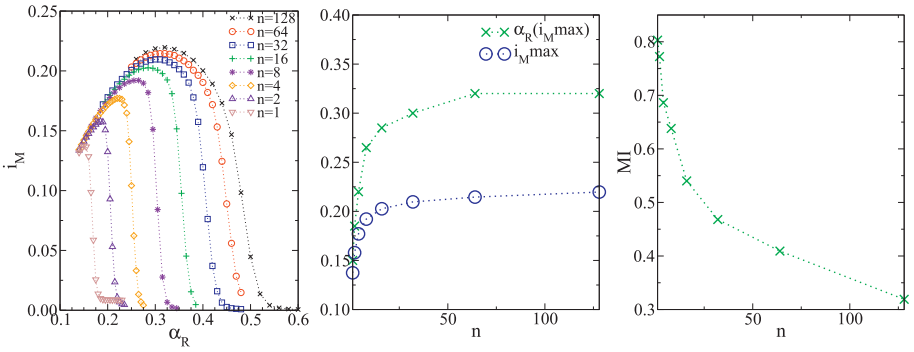

Fig. 6. Left: Information ratio i_M for the different values of the number of components in the ensemble n. Middle: Value of α_R for maximal (i_M). Right: Mutual Information (MI) vs. number of components in the ensemble n. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

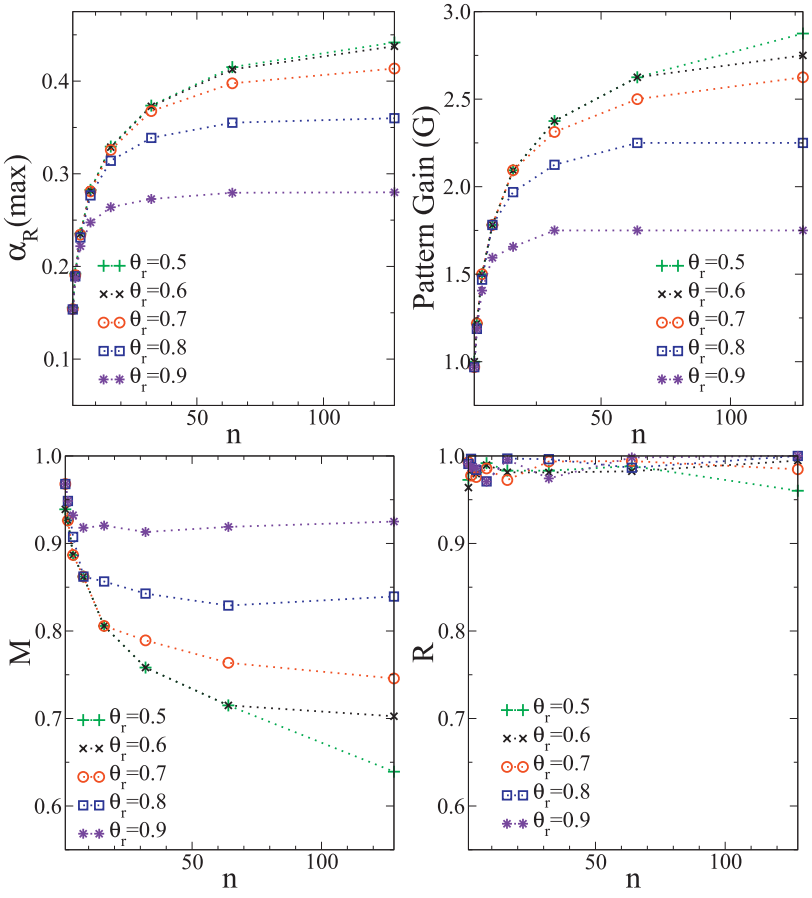

Fig. 7. ANN ensemble performance for different values of retrieval threshold θ_r. Top-left panel: maximum retrieval load α_R(max). Top-right panel: pattern gain G. Bottom-left panel: mean retrieval overlap M. Bottom-right panel: retrieval efficiency R. Increasing the components n in the ensemble leads to better ensemble performance α_R, G for all values of θ_r, while keeping the retrieval efficiency R high, despite the decrease of the retrieval quality M. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ter performance in terms of α_R(max), and pattern gain G, at high levels of retrieval efficiency R despite the decrease of the retrieval quality M, as shown in Fig. 7.

4. Conclusions

We have proposed an ensemble model of diluted ANN components that increases the number of patterns that can be processed by Hopfield-type networks. By keeping the same computational cost of wiring when compared with a single ANN system, the ANN ensemble is capable of almost triple the storage capacity of the single ANN system. Given that the transition from retrieval to non retrieval phases is continuous, in terms of the retrieval overlap m, a trade-off occurs in the proposed system. The patterns are allowed to be retrieved with some level of noise $m^\mu < 1$ as long as these are clearly discriminated from patterns in different subsets assigned to different components in the ensemble.

We tested the ANN ensemble for different levels of component connectivity K_b, and different number of components n, keeping the same wiring cost (number of total connections) in the ANN ensemble as in the single "big" ANN used as reference, $K = n \times K_b = 6400$ and for a network size $N = 10^4$. The retrieval pattern load α_R and pattern gain G were measured to determine the performance of the ensemble. Also the retrieval efficiency R and the mean retrieval overlap M were measured for each system. For an ensemble with a small number of ANN components ($K_b = 3200, n = 2$) the

retrieval pattern load is $\alpha_R \sim 0.19$ with a high mean retrieval overlap $M = 0.93$, and a pattern gain of $G \sim 24\%$. By using a larger ensemble of very diluted components ($K_b = 50, n = 128$) and with a mean retrieval overlap of $M \sim 0.64$, we can increase the number of patterns that can be stored to $\alpha_R \sim 0.44$, as shown in Section 3.2. The larger the number of components is, the larger the pattern gain of the ensemble system is, $G = 2.875$ for $K_b = 50, n = 128$. And, consequently this is also true for the maximum pattern load α_R that can be achieved by the ANN ensemble (see Section 3.3). We also calculated the information ratio i_M of the ensemble in terms of the mean retrieval overlap M, and the retrieved pattern load α_R, showing that the information ratio increases with the number of components n in the ensemble. It is worthwhile to mention that the behavior of the ensemble retrieval abilities is similar for increasing values of θ_r, the larger the number of components n in the ensemble the larger the capacity of the system (as shown in Section 3.5).

The performance increase of the proposed ensemble system can be useful to deal with real world applications where an extensive number of patterns must be processed. The suggested divide-and-conquer approach can be seen as a specialization of each network/component in the ensemble to deal with specific types of patterns. We may think of a system with a given requirement of capacity in terms of memory. How can we increase the capacity of a densely connected ANN that is at the limit of storage capacity given a constraint in terms of the number of connections? By dividing the "big" network into an ensemble of very diluted ANN modules, we are able to increase the system storage capacity using the same wiring cost (number of connections). This can be valuable to approach many Engineering problems in order to store several patterns in which the technical characteristics of memory are limited like embedded systems and smart-phones.

Also, by dividing problems into subproblems, we may overcome the cross-correlation between patterns, and deal with real data where information is structured and correlated such as fingerprints pattern recognition problems (Gonzalez et al., 2014). We could also consider non homogeneous ANN modules in terms of the connectivity degree, having denser ANN components, for generalization, and very diluted ANN components, for specialization according to disjoint features of the patterns in the set. This is related with the trade-off between specialization and generalization which is a matter of concern in machine learning, as demonstrated in the work of Barra, Bernacchia, Santucci, and Contucci (2012), where they mapped a Hopfield network to a restricted Boltzmann Machine being able to construct large networks with low storage capacity, and on the other extrema networks with a high storage regime (high number of stable states), defining a technique to find the optimal trade-off between flexibility and generalization. Also there are several works in this context (Montero, Huerta, and Rodr'ıguez (2014); Montero, Huerta, and Rodriguez, 2015a, 2015b) which investigate the optimal balance between neuronal generalism and specialism in the process of pattern recognition in biological inspired networks. In future work, we will also explore processing structured and correlated information in order to test the performance of the proposed ensemble system when dealing with this type of patterns.

Acknowledgment

This work was funded by UNEMI-2016-CONV-P-01-01, and by Spanish projects of Ministerio de Economía y Competitividad TIN-2010-19607, TIN2014-54580-R, TIN2014-57458-R (http://www.mineco.gob.es/). The funders had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript. We thank the Reviewers of the paper. Their thoughtful review and valuable comments and suggestions have been very helpful to improve this work.

References

Agliari, E., Barra, A., Galluzzi, A., Guerra, F., & Moauro, F. (2012). Multitasking associative networks. *Physical Review Letters, 109*, 268101.

Agliari, E., Barra, A., Galluzzi, A., Guerra, F., Tantari, D., & Tavani, F. (2015a). Hierarchical neural networks perform both serial and parallel processing. *Neural Networks, 66*, 22–35.

Agliari, E., Barra, A., Galluzzi, A., Guerra, F., Tantari, D., & Tavani, F. (2015b). Metastable states in the hierarchical Dyson model drive parallel processing in the hierarchical Hopfield network. *Journal of Physics A: Mathematical and Theoretical, 48*, 015001. http://stacks.iop.org/1751-8121/48/i=1/a=015001.

Agliari, E., Barra, A., Galluzzi, A., Guerra, F., Tantari, D., & Tavani, F. (2015c). Retrieval capabilities of hierarchical networks: From Dyson to Hopfield. *Physical Review Letters, 114*, 028103.

Amit, D. J. (1989). *Modeling brain function: The world of attractor neural networks*. New York, NY, USA: Cambridge University Press.

Amit, D. J., Gutfreund, H., & Sompolinsky, H. (1987). Information storage in neural networks with low levels of activity. *Physical Review A, 35*, 2293–2303. doi:10.1103/PhysRevA.35.2293.

Amit, Y., & Mascaro, M. (1999). Attractor networks for shape recognition. *Neural Computation, 13*, 1415–1442.

Arenzon, J. J., & Lemke, N. (1994). Simulating highly diluted neural networks. *Journal of Physics A: Mathematical and General, 27*, 5161. http://stacks.iop.org/0305-4470/27/i=15/a=016.

Barra, A., Bernacchia, A., Santucci, E., & Contucci, P. (2012). On the equivalence of Hopfield networks and Boltzmann machines. *Neural Networks, 34*, 1–9.

Bhagat, S., & Deodhare, D. (2006). Divide and conquer strategies for MLP training. In *International joint conference on neural networks, 2006. IJCNN'06.* (pp. 3415–3420). IEEE.

Brunel, N. (1993). Effect of synapse dilution on the memory retrieval in structured attractor neural networks. *Journal de Physique I, 3*, 1693–1715.

Brunel, N. (2003). Dynamics and plasticity of stimulus-selective persistent activity in cortical network models. *Cerebral Cortex, 13*, 1151–1161.

Derrida, B., Gardner, E., & Zippelius, A. (1987). An exactly solvable asymmetric neural network model. *Europhysics Letters, 4*, 167–173.

Dominguez, D., & Bollé, D. (1998). Self-control in sparsely coded networks. *Physical Review Letters, 80*, 2961.

Dominguez, D., González, M., Rodríguez, F. B., Serrano, E., E. , J. R., & Theumann, W. (2012). Structured information in sparse-code metric neural networks. *Physica A: Statistical Mechanics and its Applications, 391*, 799–808. doi:10.1016/j.physa.2011.09.002. http://www.sciencedirect.com/science/article/pii/S0378437111007187.

Dominguez, D., González, M., Serrano, E., & Rodríguez, F. B. (2009). Structured information in small-world neural networks. *Physical Review E, 79*, 021909. doi:10.1103/PhysRevE.79.021909.

Dominguez, D., Koroutchev, K., Serrano, E., & Rodríguez, F. B. (2007). Information and topology in attractor neural network. *Neural Computation, 19*(4), 956–973.

Erdös, P., & Rényi, A. (1959). On random graphs, I. *Publicationes Mathematicae (Debrecen), 6*, 290–297. http://www.renyi.hu/~p_erdos/Erdos.html#1959-11.

Fink, W. (2004). Neural attractor network for application in visual field data classification. *Physics in Medicine and Biology, 49*, 2799. http://stacks.iop.org/0031-9155/49/i=13/a=003.

González, M., Dominguez, D., & Rodríguez, F. B. (2009). Block attractor in spatially organized neural networks. *Neurocomputing, 72*, 3795–3801. doi:10.1016/j.neucom.2009.05.010.

Gonzalez, M., Dominguez, D., Rodriguez, F. B., & Sanchez, A. (2014). Retrieval of noisy fingerprint patterns using metric attractor networks. *International Journal of Neural Systems, 24*, 1450025.

González, M., Dominguez, D., & Sánchez, A. (2011). Learning sequences of sparse correlated patterns using small-world attractor neural networks: An application to traffic videos. *Neurocomputing, 74*, 2361–2367. doi:10.1016/j.neucom.2011.03.014. http://www.sciencedirect.com/science/article/pii/S092523121100155X.

Hertz, J., Krogh, J., & Palmer, R. (1991). *Introduction to the theory of neural computation*. Boston: Addison-Wesley.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America, 79*, 2554–2558. http://www.pnas.org/cgi/content/abstract/79/8/2554.

Johansson, C., Rehn, M., & Lansner, A. (2006). Attractor neural networks with patchy connectivity. *Neurocomputing, 69*, 627–633.

Lopez, M., Melin, P., & Castillo, O. (2007). A method for creating ensemble neural networks using a sampling data approach. In P. Melin, O. Castillo, E. R. Arez, J. Kacprzyk, & W. Pedrycz (Eds.), *Analysis and design of intelligent systems using soft computing techniques*. In *Advances in Soft Computing: 41* (pp. 355–364). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-72432-2_36.

Montero, A., Huerta, R., & Rodríguez, F. B. (2014). Neural trade-offs among specialist and generalist neurons in pattern recognition. In *International conference on engineering applications of neural networks* (pp. 71–80). Springer.

Montero, A., Huerta, R., & Rodriguez, F. B. (2015a). Regulation of specialists and generalists by neural variability improves pattern recognition performance. *Neurocomputing, 151*, 69–77.

Montero, A., Huerta, R., & Rodriguez, F. B. (2015b). Specialist neurons in feature extraction are responsible for pattern recognition process in insect olfaction. In *International work-conference on the interplay between natural and artificial computation* (pp. 58–67). Springer.

Noel, S., & Szu, H. (1997). Multiple-resolution divide and conquer neural networks for large-scale TSP-like energy minimization problems. In *International conference on neural networks: 2* (pp. 1278–1283). doi:10.1109/ICNN.1997.616218.

Rolls, E. T., & Webb, T. J. (2012). Cortical attractor network dynamics with diluted connectivity. *Brain Research, 1434*, 212–225.

Roudi, Y., & Treves, A. (2008). Representing where along with what information in a model of a cortical patch. *PLoS Computational Biology, 4*, e1000012. doi:10.1371/journal.pcbi.1000012.

Ruppin, E., & Yeshurun, Y. (1991). Recall and recognition in an attractor neural network model of memory retrieval. *Connection Science, 3*, 381–400.

Sollich, P., Tantari, D., Annibale, A., & Barra, A. (2014). Extensive parallel processing on scale-free networks. *Physical Review Letters, 113*, 238106.

Sompolinsky, H. (1986). Neural networks with nonlinear synapses and a static noise. *Physical Review A, 34*, 2571.

Stringer, S. M., Rolls, E. T., Trappenberg, T. P., & de Araujo, I. E. T. (2003). Self-organizing continuous attractor networks and motor function. *Neural Networks, 16*, 161–182. doi:10.1016/S0893-6080(02)00237-X. http://dl.acm.org/citation.cfm?id=781392.781394.

Wang, X., Tegner, J., & Constantinidis, C. (2004). Division of labor among distinct subtypes of inhibitory neurons in a cortical microcircuit of working memory. *Proceedings of the National Academy of Sciences, 101*, 1368–1373. http://www.pnas.org/cgi/content/abstract/101/5/1368.

Wemmenhove, B., & Coolen, A. (2003). Finite connectivity attractor neural networks. *Journal of Physics A: Mathematical and General, 36*, 9617.

Wills, T. J., Lever, C., Cacucci, F., Burgess, N., & OKeefe, J. (2005). Attractor dynamics in the hippocampal representation of the local environment. *Science, 308*, 873–876.