frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 10 January 2017
doi: 10.3389/fncom.2016.00144

On the Maximum Storage Capacity of the Hopfield Model

Viola Folli[1]*, *Marco Leonetti*[1] and *Giancarlo Ruocco*[1, 2]

[1] Center for Life Nanoscience, Istituto Italiano di Tecnologia, Rome, Italy, [2] Department of Physics, Sapienza University of Rome, Rome, Italy

Recurrent neural networks (RNN) have traditionally been of great interest for their capacity to store memories. In past years, several works have been devoted to determine the maximum storage capacity of RNN, especially for the case of the Hopfield network, the most popular kind of RNN. Analyzing the thermodynamic limit of the statistical properties of the Hamiltonian corresponding to the Hopfield neural network, it has been shown in the literature that the retrieval errors diverge when the number of stored memory patterns (P) exceeds a fraction ($\approx 14\%$) of the network size N. In this paper, we study the storage performance of a generalized Hopfield model, where the diagonal elements of the connection matrix are allowed to be different from zero. We investigate this model at finite N. We give an analytical expression for the number of retrieval errors and show that, by increasing the number of stored patterns over a certain threshold, the errors start to decrease and reach values below unit for $P \gg N$. We demonstrate that the strongest trade-off between efficiency and effectiveness relies on the number of patterns (P) that are stored in the network by appropriately fixing the connection weights. When $P \gg N$ and the diagonal elements of the adjacency matrix are not forced to be zero, the optimal storage capacity is obtained with a number of stored memories much larger than previously reported. This theory paves the way to the design of RNN with high storage capacity and able to retrieve the desired pattern without distortions.

Keywords: maximum storage memory, feed-forward structure, random recurrent network, Hopfield model, retrieval error

OPEN ACCESS

Edited by:
Marcel Van Gerven, Radboud University Nijmegen, Netherlands

Reviewed by:
Simon R. Schultz, Imperial College London, UK
Fleur Zeldenrust, Donders Institute for Brain, Cognition and Behaviour, Netherlands

***Correspondence:**
Viola Folli
viola.folli@iit.it

Received: *19 September 2016*
Accepted: *20 December 2016*
Published: *10 January 2017*

Citation:
Folli V, Leonetti M and Ruocco G (2017) On the Maximum Storage Capacity of the Hopfield Model. Front. Comput. Neurosci. 10:144. doi: 10.3389/fncom.2016.00144

1. INTRODUCTION

A vast amount of literature deals with neural networks, both as model for brain functioning (Amit, 1989), and as smart artificial systems for practical applications in computation and information handling (Haykin, 1999).

Among the different possible applications of artificial neural networks, those referred to as "associative memory" are particularly important (Rojas, 1996), i.e., circuits with the capability to store and retrieve specific information patterns. According to Amit et al. (1985a,b) there is a natural limit for the usage of an N nodes neural network built according to the Hebbian principle (Hebb, 1949) as associative memory. The association is embedded within the connection matrix which has a dyadic form: the weight connecting neuron i to neuron j is the product of the respective signals. The limit of storage is linear with N: an attempt to store a number P of memory elements larger than $\alpha_c N$, with $\alpha_c \approx 0.14$, results in a "divergent" (order P) number of retrieval errors. In order to be effective (low retrieval error probability) a neural network working as associative memory cannot be efficient (i.e., it can store only a small number of memory elements). This is particularly frustrating in practical applications, as it strongly limits the use of artificial neural networks for information

storage, especially since it is well known that the number of fixed points in randomly connected (symmetric) neural networks shows an exponential relation with N (Tanaka and Edwards, 1980; Sompolinsky et al., 1988; Wainrib and Touboul, 2013).

Contemporaneous to Amit et al., Abu-Mostafa, and St. Jaques (Abu-Mostafa et al., 1985) claimed that the number of fixed points that can be used for memory storage in a Hopfield model with a generic coupling matrix is limited to N (i.e., $P<N$). Soon after, Mc Eliece et al. (1987), considering only the Hebbian dyadic form for the coupling matrix, found a more severe limitation: the maximum P scales as $N/\log(N)$. In a more recent study, Sollacher et al. (2009) designed a network of specific topology, reaching α_c-values larger than 0.14, but still maintaining the limit of a linear N dependence of the maximum storage capacity. The storage problem remains an open research question (Brunel, 2016).

In this letter we show that the existence of a critical P/N-value in the Hebbian scheme for the coupling matrix is only part of the story. As demonstrated in Amit et al. (1985a,b), the limit $P<\alpha_c N$ holds in the region where $P<N$. In all previous studies, the diagonal elements are removed from the dyadic form of the coupling matrix. Here we show the existence of a not yet explored region in the parameter space, with $P \gg N$, where the number of retrieval errors decreases with increasing P and reaches values lower than one. This region can be found by not removing the diagonal elements. Strictly speaking the present model is not a "Hopfield model," as in the latter case the diagonal elements are forced to vanish and—as we will see- bring significant differences in the network behavior. In order to avoid confusion, let us call the present model as "Hopfield model with autapses" or "Generalized Hopfield model." This strategy allows the design of effective and efficient associative memories based on artificial neural networks. In the following we will derive analytically the probability of retrieval errors, validate these results by their comparison with a numerical simulation and study the efficiency of the system as a function of P and N.

2. METHODS

2.1. Network Model

In an artificial neural network working as associative memory, one deals with a network of N neurons of which each one has state s_i ($i = 1...N$) that can be "active" ($s_i = 1$) or "quiescent" ($s_i = -1$). The configuration of the whole network is given by the vector $\bar{s} \equiv \{s_1, s_2, ..s_N\}$ and its temporal evolution follows the parallel non-linear dynamics:

$$s_i(t + \Delta t) = E[s_i(t)] \doteq \text{sign}\Big[\sum_{j=1}^{N} J_{ij} s_j(t)\Big], \tag{1}$$

where $\mathbf{J} = \{J_{ij}\}$ is the connection matrix. We set external inputs to be equal to 0. We assume a symmetric bimodal distribution for the synaptic polarities in the wiring matrix $\mathbf{J}$, so 50% of the connections are excitatory and 50% inhibitory. After a transient time related to the finite value of N, the network reaches a fixed point, $s_i = E[s_i]$, or a limit cycle of length L, $s_i = E^{(L)}[s_i]$.

2.2. The Hebbian Rule and the Storage Memory

Previous work has studied the cycle length and transient time distribution as a function of the properties of $\mathbf{J}$ (Gutfreundt et al., 1988; Sompolinsky et al., 1988; Derrida, 1989; Bastolla et al., 1997). In order to work as an associative memory, the matrix $\mathbf{J}$ must be tailored in such a way that one or more patterns of neurons are fixed points of the dynamics in Equation (1), i.e., they are the "memory elements" stored in the network. To store one pattern $\bar{\xi}$, the connection matrix is simply the dyadic form given by $J_{ij} = \xi_i \xi_j$ [1], while to store a generic number P of patterns $\bar{\xi}^\mu$ ($\mu = 1...P$) one follows the storage prescription of Cooper (1973) and Cooper et al. (1979), who exploited an old idea which goes back to Hebb (1949) and Eccles (1953) and which states that the change in synaptic transmission is proportional to the product of the signals of pre and post-synaptic neurons. The process for which each matrix element is appropriately determined is called *learning*. Specifically, the "Hebbian" rule results in the following expression for the connectivity matrix,

$$J_{ij} = \frac{1}{P} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu. \tag{2}$$

The set of vectors $\bar{\xi}^m u$ is known as "training set." In this case, it is not guaranteed that each $\bar{\xi}^\mu$ is a fixed point. In other words, $\bar{\xi}^\mu$ is stable in probabilistic sense. Further, the probability for $\bar{\xi}^\mu$ to be a fixed point depends on the values of P and N. This dependence has been first studied by Hopfield (1982); Hopfield et al. (1983); Hopfield (1984) who concluded that the retrieval of the memory stored in the Hebbian matrices is guaranteed up to a P-value which is a critical fraction on the number of network nodes N of the order of 10–20%. Above this value, the associative memory quickly degrades. Following these studies, Amit et al. (1985a,b), who noticed the similarity between the Hopfield model for the associative memory and the spin glasses, developed a statistical theory for the determination of the critical P/N ratio, that turned out to be ≈ 0.14, in good agreement with the previous Hopfield estimation. Above P=0.14N the number of errors is so large that the network based on the Hebbian matrix is no longer capable to work as an associative memory. All these studies assumed a modified form of Equation (2): the diagonal elements of $\mathbf{J}$ are forced to be zero.

2.3. Numerical Simulations and Data Analysis

In order to demonstrate the validity of our analytic results (see Section 3), we perform numerical simulations by evolving the network model as described in Equation (1). We design the default network by fixing the NxN recurrent connections as given in Equation (2), by randomly assigning the value ± 1 to ξ_i^μ and retaining the diagonal elements. So, the $N(N-1)$ connections are 50% excitatory and 50% inhibitory and the N neurons can form self-connections. We then run simulations by varying the size of the network, $N = 50, .., 200$ and the number of stored memories

[1] The evolution $E[\xi_i]$ always return ξ_i since the sum $\sum_j \xi_j^2 = N$.

in Equation (2), $P = 1, ..., 2000$. Finally, for each pair of N and P, we perform 1000 different random realizations.

All P patterns introduced in Equation (2) are given as input to the network and their dynamics is followed until the network reaches the equilibrium state. The initial patterns are chosen among those that were stored in the adjacency matrix and that have been randomly chosen in the designing of the network. Evolved patterns were recorded at each time step and compared with the initial one. Then, if the evolved pattern is different from the initial state, we calculated the temporal evolution of the distortion (number of wrong bits) and determined the probability that one of the bits was wrong, the probability that the whole vector was exactly recovered, and the number of memory patterns that could not be recovered, as a function of N and P. Basically, to calculate the storage capacity, it is sufficient to determine all these quantities by using the distortion between the stimulus (the stored memory) and its first evolved pattern.

3. RESULTS

3.1. The Probability of Recovery

In order to investigate the maximum storage memory of our model, we calculate the one-step dynamical evolution. We give as input a vector of the training set and we calculate a single step of the dynamical evolution according to Equation (1). Then, we compare the output with the input. We aim to look whether or not a vector, $\bar{\xi}^{\mu}$, belonging to the training set, is truly a fixed point. If $\bar{\xi}^{\mu}$ is a fixed point, the output coincides with the input, and the recovery has been successful. If $\bar{\xi}^{\mu}$ is not a fixed point, the two vectors differ for at least a single bit. We now derive an analytical expression for the probability that the recovery of a stored pattern was not successful. The first step is to find the probability p_B that -given the matrix $\mathbf{J}$ of Equation (2)- a single element of the vector (a "bit") was wrong, i.e., the probability that $E[\xi_i^{\mu}] \neq \xi_i^{\mu}$. Basically, we need to evolve a vector $\bar{\xi}^m u$ (from the training set) for one step and count how many bits of its time evolution are different from the bits of $\bar{\xi}^m u$ itself. Obviously, if $\bar{\xi}^m u$ actually is a fixed point, this distance vanishes. On the contrary, $\bar{\xi}^m u$ is not a fixed point, the network has made a recovery error. Thus, p_B (or better, p_V, as we see in the next paragraph) measures "how many" training set vectors are not fixed points. The argument of the *sign* function in Equation (1) is $A_i^{\mu} = \sum_{j=1}^{N} \sum_{\nu=1}^{P} \xi_i^{\nu} \xi_j^{\nu} \xi_j^{\mu}$, this contains NP terms among which there are $N+P-1$ terms (those with $j=i$ [2] and those with $\nu=\mu$) where two out of the three ξ of the product are equals to each other ξ_i^{ν} and the third is ξ_i^{μ}. Thus $A_i^{\mu} = (N+P-1)\xi_i^{\mu} + T_i^{\mu}$, with $T_i^{\mu} = \sum_{j\neq i}^{N} \sum_{\nu\neq\mu}^{P} \xi_i^{\nu} \xi_j^{\nu} \xi_j^{\mu}$. The first term is the "coherent" one, its sign is identical to ξ_i^{μ}, and it will -if dominant- guarantee that $\bar{\xi}^{\mu}$ is a fixed point of the dynamics. The second term T_i^{μ}, on the contrary, is "noise" and its presence can either reinforce or weaken the stability of ξ_i^{μ} as fixed point. Specifically, if $|T_i^{\mu}| > (N+P-1)$ and $\text{sign}(T_i^{\mu}) \neq \xi_i^{\mu}$, then the *i-th* bit of the vector $\bar{\xi}^{\mu}$ will turn out to be wrong. The quantity T_i^{μ} is the sum of $(N-1)(P-1)$ statistically independent terms, each one being $+1$ or -1. Therefore, for large enough P and N, its distribution $N(T)$ can be approximated by a gaussian with zero mean and standard deviation $\sqrt{(N-1)(P-1)}$:

$$N(T) = \frac{e^{-T^2/(2(N-1)(P-1))}}{\sqrt{2\pi(N-1)(P-1)}}. \tag{3}$$

It is now straightforward to determine the probability that $|T_i^{\mu}| > (N+P-1)$ and $\text{sign}(T_i^{\mu}) \neq \xi_i^{\mu}$, thus that one of the bits of $E[\xi_i^{\mu}]$ was wrong, as $p_B = \int_{N+P-1}^{\infty} dT\, N(T)$. In conclusion:

$$p_B = \frac{1}{2}\Big[1 - \text{erf}\Big(\frac{N+P-1}{\sqrt{2(N-1)(P-1)}}\Big)\Big]. \tag{4}$$

It is worth to note that this expression is symmetric under the exchange of P with N, and that for large P and N, with $P/N = 1$, it tends to $(1-\text{erf}(\sqrt{2}))/2 \approx 0.02275$ which corresponds to the maximum of probability in a wrong recovery of a single bit (see **Figures 1**, **2**).

The second step is the determination of the probability p_V that one of the P vectors encoded into the connection matrix (the training set) turns out not be a fixed point. If only a single bit of the vector is wrong, the whole vector is considered "wrong." Since there are N bits that can be wrong, the probability p_V will be much higher than p_B. The calculation is straightforward, in order not to be wrong, all the bits of the vector $\bar{\xi}^{\mu}$ must be right, thus $p_V = 1-(1-p_B)^N$, therefore:

$$p_V = 1 - \Big[\frac{1}{2} + \frac{1}{2}\text{erf}\Big(\frac{N+P-1}{\sqrt{2(N-1)(P-1)}}\Big)\Big]^N. \tag{5}$$

Finally, the number, N_V, of memory vectors that are not recovered, i.e., that are not true fixed points of the dynamics is given by Pp_V, that is:

$$N_V = \Big[1 - \Big[\frac{1}{2} + \frac{1}{2}\text{erf}\Big(\frac{N+P-1}{\sqrt{2(N-1)(P-1)}}\Big)\Big]^N\Big]P \tag{6}$$

3.2. The Asymptotical Approximation

Equations (4), (5) and, in particular, Equation (6) represent the main result of this work. Before showing their validity, via a comparison with numerical simulations, and discussing their relevance in the framework of artificial neural networks, it is important to present the asymptotical approximation for N_V. The argument of the error function, for either $P \gg N$ or $P \ll N$, is large, and can be expanded as $\text{erf}(x) \approx 1-\exp(-x^2)/(x\sqrt{\pi})$. Furthermore, as p_B is exponentially small with N (or P) for large N (P), we use, $(1-p_B)^N \approx (1-Np_B)$. Thus, for large N or large P:

$$p_V \approx \frac{N^{3/2}P^{1/2}e^{-\frac{(N+P)^2}{2NP}}}{\sqrt{2\pi}(N+P)} \tag{7}$$

$$N_V \approx \frac{N^{3/2}P^{3/2}e^{-\frac{(N+P)^2}{2NP}}}{\sqrt{2\pi}(N+P)} \tag{8}$$

We note that, while in the exact expression for N_V (Equation 6) the $P \leftrightarrow N$ exchange symmetry is lost, in the approximate form the symmetry is recovered.

[2] These are P terms that are present only if the diagonal elements are kept as they are and are not forced to vanish.

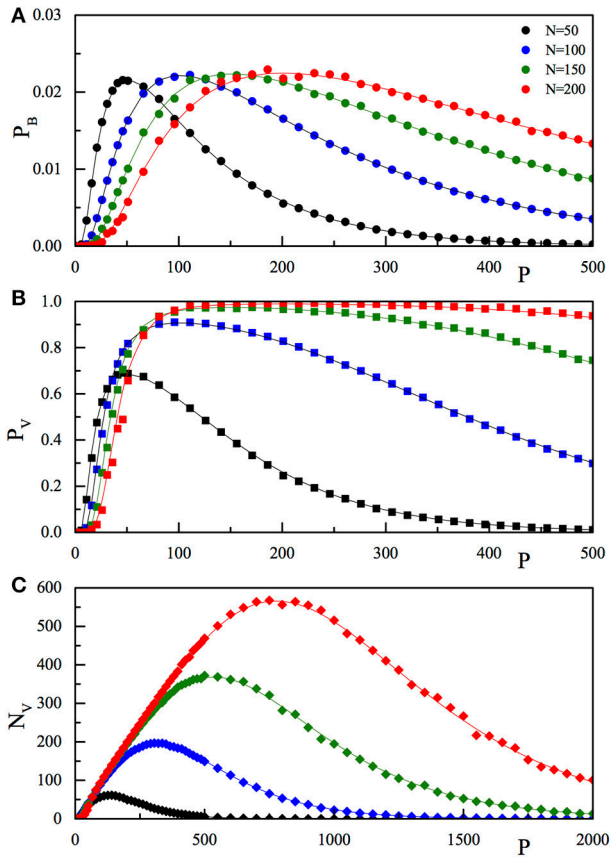

FIGURE 1 | Comparison of the results of the numerical simulation (p_B, full dots A; p_V, full squares, **B**; N_V, full diamonds, **C**) with the corresponding theoretical function (p_B, Equation 4; p_V, Equation 5; N_V, Equation 6) reported as full lines. The three quantities are reported as a function of P for fixed N. The values of N are 50 (black), 100 (blue), 150 (green), and 200 (red). The P range in **(C)** is extended with respect to **(A,B)**.

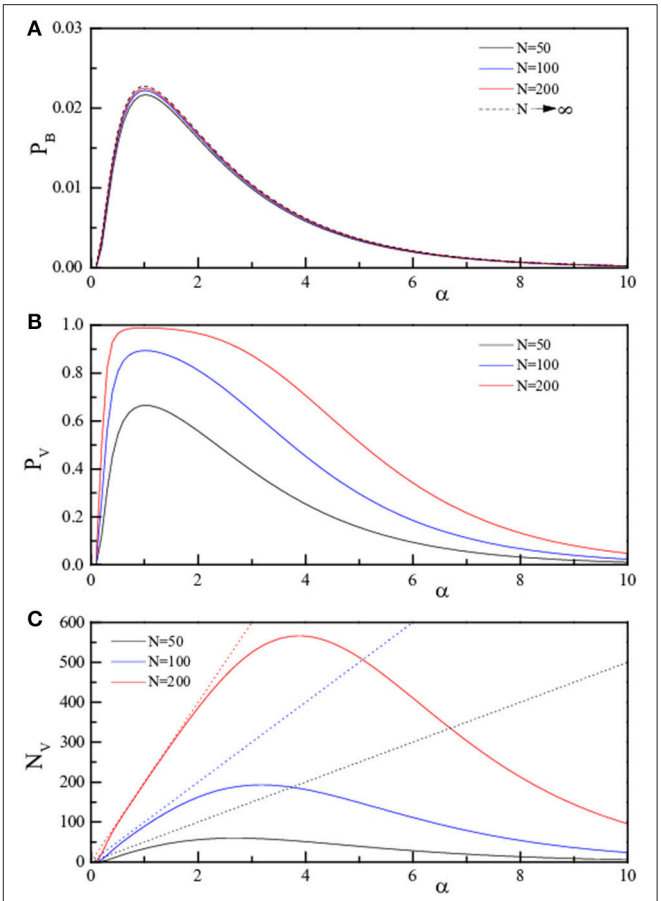

FIGURE 2 | Theoretical curves for the three quantities p_B, p_V, and N_V (p_B, Equation 4, A; p_V, Equation 5, **B**; N_V, Equation 6, **C**) reported as full lines. The three quantities are reported in linear scale as a function of $\alpha = P/N$ for fixed N. The values of N are 50 (black), 100 (blue), and 200 (red). The dotted lines in **(C)** represent $N_V = P$.

For sake of comparison with the previous literature, it is also useful to express the main results as a function of $\alpha \doteq P/N$. Equations (4) (for large N) and (8) read:

$$p_B \approx \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{1+\alpha}{\sqrt{2\alpha}}\right)\right] \tag{9}$$

$$N_V \approx NP\frac{1}{\sqrt{2\pi}}\frac{\sqrt{\alpha}}{1+\alpha}e^{-\frac{(1+\alpha)^2}{2\alpha}}. \tag{10}$$

While p_B only depends on α, N_V clearly is an extensive observable, being proportional to P and N. Furthermore, both expressions keep their symmetry with respect to the exchange of P and N, thus to the exchange of α with $1/\alpha$. The last observation anticipates that there must exists a region at large α-values where the same features are observed as at small values of α.

3.3. Numerical Results

To check the predictions of our network model, we have simulated the Model (1) and studied the dynamics for several values of N and P, in the range of few hundred, see Section 2.3 for details. In the numerical analysis, the P memory vectors have been randomly chosen and used to construct the connection matrix **J**. Next, we tested whether or not the stored memories were fixed points of the dynamics. The values of p_B, p_V and N_V were calculated by averaging over (up to) 1000 different random realizations of $\bar{\xi}^\mu$. The results of the numerical simulations are reported (dots) in **Figure 1**, together with the analytical Expressions (4)–(6) (lines). The three panels refer to the three quantities p_B (**Figure 1A**), p_V (**Figure 1B**), and N_V (**Figure 1C**) as a function of P for the selected values of N, as reported in the legend. From **Figure 1**, we observe that on increasing P, at fixed N, both the single bit probability error, the probability of recovery error (P_V), and the number of wrong recoveries N_V, after a first fast increase, reach a maximum (equal to 0.02275 for p_B, close to one for p_V, and larger than N for N_V) then start to decrease, tending to zero for very large P-values.

To better emphasize this behavior, the same quantities are reported (analytic results only) as a function of α in **Figure 2** (linear scale) and in **Figure 3** (log scale) for selected N. The dotted lines in panels C of both figures represent $N_V = P$,

i.e., indicate the case of "totally wrong recovery." Due to the already observed $\alpha \leftrightarrow 1/\alpha$ symmetry, the asymptotic curve in **Figure 3A** appears with a left-right symmetry around $\alpha = 1$. From **Figures 2**, **3**, we can clearly identify two regions of high recovery efficiency. The low α region, already studied many years ago by Hopfield (1982); Hopfield et al. (1983); Hopfield (1984) and Amit et al. (1985a,b), shows the existence of a quick transition toward "loss of memory recover" on increasing α around $\alpha \approx 0.14$. The second region at large α-values is not yet explored.

Although the value $\alpha = 1$ ($P = N$) represents traditionally a sort of limit in the computation of the storable memories in a RNN, there is no reason why not to store more than N memory elements in a network of N neurons, that by construction allows 2^N possible patterns. Indeed, the number of fixed points in a (random) symmetric matrix is known to be, for fully connected symmetric matrices as in our case, exponentially large with N (Tanaka and Edwards, 1980). Specifically, the number of fixed points P_o is equal to $P_o = \exp(\gamma N)$, with $\gamma \approx 0.2$. P_o, much larger than N, can be considered a natural limit for P.

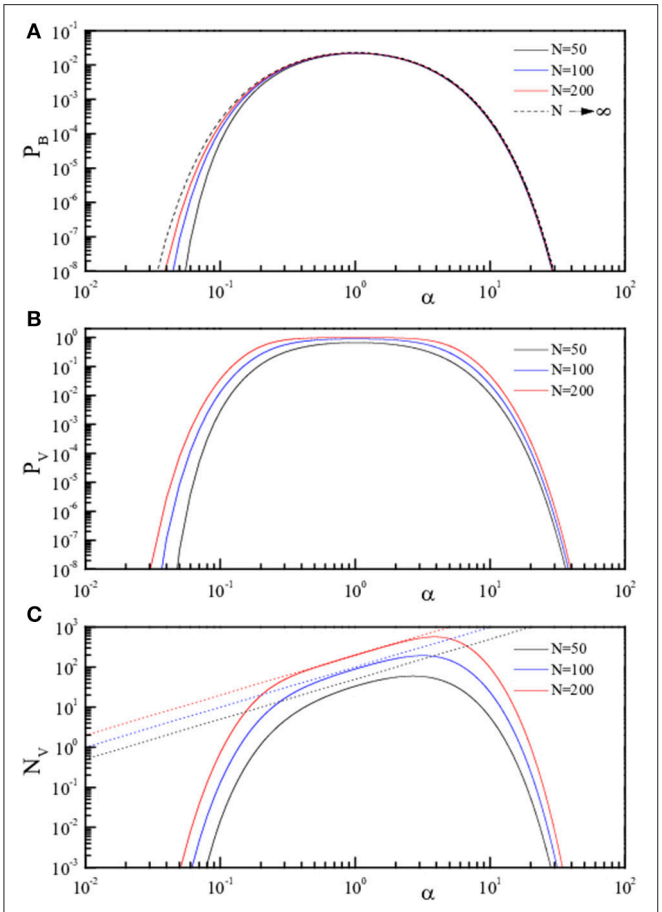

FIGURE 3 | Theoretical curves for the three quantities *pB*, *pV*, and *NV* (*pB*, Equation 4, A; *pV*, Equation 5, **B**; *NV*, Equation 6, **C**) reported in Log-Log scale as full lines. The three quantities are reported in linear scale as a function of $\alpha = P/N$ for fixed N. The values of N are 50 (black), 100 (blue), and 200 (red). The dotted lines in **(C)** represent $NV = P$.

The recovery efficiency increases for large P. In fact, the coherent term in the argument of the sign function increases linearly with P and the noise increases as $P^{1/2}$. For large P, the relative weight of the noise decreases as $P^{-1/2}$, this allows to store a large number of memories in a relatively small neural network.

For practical purposes, as for example in the design of an artificial neural network with high efficiency (large storage capacity) and effectiveness (low recovery error rates), it is important to study (Equation 6, and its approximation in Equation 8) and, in particular, to find the conditions for which the network shows "perfect recovery." Let's define perfect recovery as the state where the number of retrieval errors N_V is smaller than one.

In **Figure 4** we show the contour plot of the (decimal) logarithm of N_V, from Equations (6) and (8), in the P-N range [0–100]. The full lines are the loci of the points where $log_{10}(N_V)$ equals 0, 0.4, 0.8, 1.2, and 1.6, as indicated on the right side of the figure. The dashed lines are the same level lines for the (logarithm of the) approximate form of N_V reported in Equation (8). As can be observed, for $N_V \approx 1$, the approximation (Equation 8) for N_V is highly accurate, indicating that this approximation can be safely applied to find the "perfect recovery" condition.

In the P-N plane the existence of two regions (small and large α) where the perfect recovery ($N_V = 1$, red lines) takes place can be easily observed and the result is symmetric under the exchange of P and N. In the already explored small α region, we also show (full blue line) the $P = 0.14N$ condition. Similar to the high α region, it is important to find a simple relation between N and P identifying the $N_V = 1$ condition. We aim, therefore, to obtain a function $P(N)$ which returns, at given N, the P-value

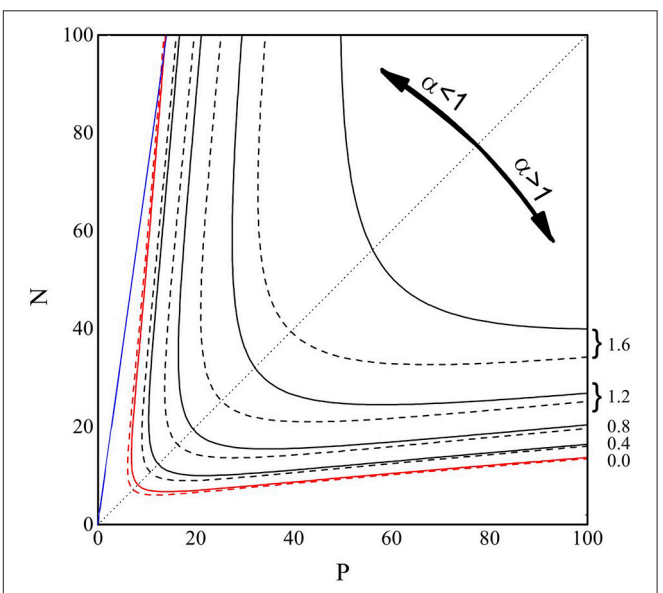

FIGURE 4 | Contour plot of $\log_{10}(N_V)$ from Equations (6) (full lines) and (8) (dashed lines), in the *P* and *N* range 0–100. The lines are the loci of the points where $log_{10}(N_V)$ equals 0.0 (red), 0.4, 0.8, 1.2, and 1.6 (black), as indicated on the right side of the figure. The blue line represents $P = 0.14N$, while the black dotted line is the bisectrix $N = P$, plotted to emphasize the symmetry of the contour lines.

such that $N_V = 1$. We write the prefactor NP in Equation (10) as αN^2 and exploit the $\alpha \gg 1$ limit, so to obtain $N_V \approx N^2\alpha^{1/2}\exp(-\alpha/2)/\sqrt{2\pi}$. The equation $N^2\alpha^{1/2}\exp(-\alpha/2)/\sqrt{2\pi} = 1$ can be squared, $\alpha\ \exp(-\alpha) = 2\pi/N^4$, and solved with respect to α, to give $\alpha = -W_{-1}(-2\pi/N^4)$, where $W_{-1}(x)$ is the second real branch of the Lambert function (Olver et al., 2010). In conclusion, the "perfect recovery condition" is satisfied -for each N-value- if we chose to store a number of memories *larger* than $P(N)$ given by:

$$P(N) = -NW_{-1}(-2\pi/N^4). \tag{11}$$

For practical purposes, for large enough N, we can use the small-argument expansion of the Lambert function $-W_{-1}(-x) \approx -\ln(x)+\ln(-\ln(x))$ (Corless et al., 1996), to have:

$$P(N) = N\left[\ln\left(\frac{N^4}{2\pi}\right) + \ln\left(\ln\left(\frac{N^4}{2\pi}\right)\right)\right]. \tag{12}$$

The results for $P(N)$ are shown in **Figure 5** as a function of N in the range 1–1000. The black line represents the exact, numerical, solution to $N_V = 1$, with N_V in Equation (6), the blue line is the expression for $P(N)$ in Equation (11), while the red line is those in Equation (12).

It is important to note that the presence of a decrease of the retrieval error probability at high P, or α, values is due to the presence of non-zero diagonal elements in the **J** matrix that creates a coherent term of weight P. Indeed, repeating the rationale leading to Equation (4) with the assumption that $J_{ii} = 0$, would give rise to the same (Equations 4–6) but with the numerator of the argument of the error functions equal to $N - 1$ instead of to $N + P - 1$. This is shown graphically in **Figure 6** where we compare for $N = 50$, both theoretically (full line) and numerically (full dots), the quantities p_B, p_V, and N_V as a function of P in the two cases: diagonal elements in Equation (2) (black) and diagonal elements forced to vanish (orange).

The stabilization of the fixed points $\bar{\xi}^\mu$ in the high storage region arises from the presence of the non-zero diagonal elements. Asymptotically, on increasing P, the diagonal elements growth coherently and the **J** matrix tends to become the unit matrix. However, the dynamics (see Equation 1) dictated by the matrix J does not tend to the dynamics dictated by the unit matrix. In the latter case, indeed, all the 2^N state vectors should become fixed points and the network should loose on important feature: the capability to distinguish between the stored memories (the vectors $\bar{\xi}^\mu$, for $\mu = 1...P$) and the spurious fixed points, all the vectors $\bar{\zeta}$ not belonging to the set $\bar{\xi}^\mu$ but such that $E[\bar{\zeta}] = \bar{\zeta}$. To study this property, we have calculated the probability that a (randomly chosen) vector $\bar{\zeta}$ (different from all the $\bar{\xi}^\mu$ used

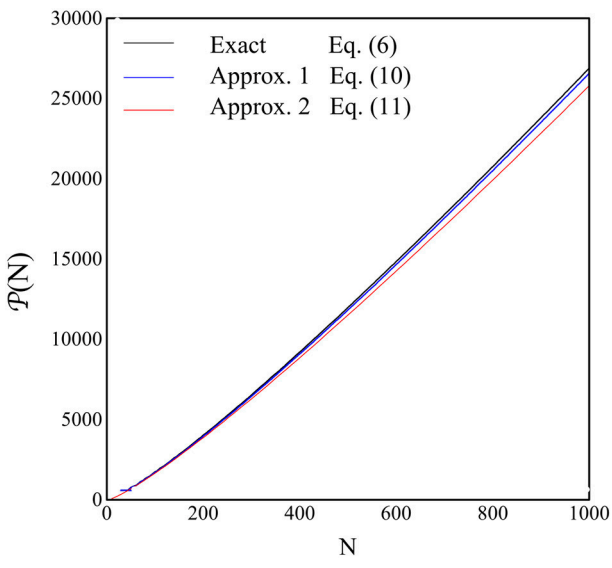

FIGURE 5 | The quantity *P*(*N*), i.e., the *P*-value where the perfect recovery is guaranteed, is shown as a function of *N*. The blue line is the numerical solution of $N_V = 1$ from Equation (6), the blue line is the plot of Equation (11) and the red line is the plot of Equation (12).

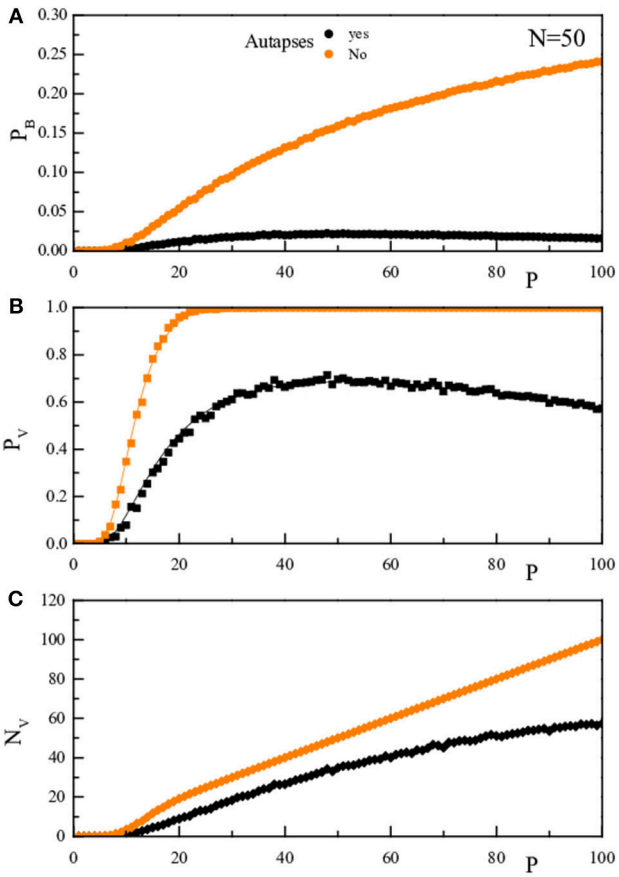

FIGURE 6 | The upper panel (A) reports for a given *N*-value (*N* = 50), as a function of P, the probability p_B that, stimulating the network with a vector inside the training set, there is one bit wrong in the network response. The middle panel **(B)** reports p_V, the probability that, stimulating the network with a vector inside the training set, the vector obtained after one dynamical step is not the stimulating vector. The lower panel **(C)** reports $N_V = Pp_V$. The black symbols/lines refer to the case where the diagonal elements are as determined in Equation (2), while the oranges ones to diagonal elements forced to vanish. The full lines are the theoretical prediction, the full dots are the results of the numerical simulation.

to build the **J** matrix) was recognized as a "memory" from the network dynamics. To be consistent with the previous notation (where we called p_B and p_V the probability of errors, not that of correct retrieval of the memory states) we define $\bar{p}_B$ ($\bar{p}_V$) as the probability of correctly not retrieving a vector not belonging to the training set. More specifically, the quantity $\bar{p}_V$ is the probability that one dynamical step after presenting a vector ζ not belonging to the training set to the network, the output a vector is different from ζ." More specifically, the quantity $\bar{p}_V$ is the probability that presenting a vector ζ not belonging to the training set to the network, after one dynamical step we found as output a vector different from ζ. Similarly for $\bar{p}_B$. It turns out that [3]:

$$\bar{p}_B = \frac{1}{2}\Big[1 - \mathrm{erf}\Big(\frac{P}{\sqrt{2(N-1)(P-1)}}\Big)\Big] \tag{13}$$

$$\bar{p}_V = 1 - \Big[\frac{1}{2} + \frac{1}{2}\mathrm{erf}\Big(\frac{P}{\sqrt{2(N-1)(P-1)}}\Big)\Big]^N. \tag{14}$$

In **Figure 7** we report the comparison of the P dependence of p_B and $\bar{p}_B$ (**Figure 7A**) and that of p_V and $\bar{p}_V$ (**Figure 7B**). As usual, full lines are the theoretical results, while the full dots are the outcome of the numerical simulation. Black data are for the "memory states," while the green ones are for the "spurious state." As can be seen, the spurious state becomes more and more "present" in the set of memories stored by the network as P increases. It seems however that also at high P-values the retrieval of the memory states is reasonably good and that of the spurious states reasonably bad.

To be quantitative on this point, we rewrite Equation (14) in its large N limit:

$$\bar{p}_V \approx \frac{N^{3/2}P^{-1/2}e^{-\frac{P}{2N}}}{\sqrt{2\pi}} \tag{15}$$

and compare it with Equation (7). In particular, is interesting to calculate the ratio, ρ, between the probability of wrong retrieval of a spurious state and that of a memory state: $\rho = \bar{p}_V/p_V$. From Equations (7) and (15) it turns out:

$$\rho = \Big(\frac{N+P}{P}\Big)\ e^{\frac{(N+P)^2}{2NP}} e^{-\frac{P}{2N}}. \tag{16}$$

This quantity only depends on α:

$$\rho = \Big(\frac{1+\alpha}{\alpha}\Big)\ e^{\left(1+\frac{1}{2\alpha}\right)} \tag{17}$$

and ρ has a finite high α limit:

$$\lim_{\alpha\to\infty} \rho = e. \tag{18}$$

In other words, although the number of spurious attractors tends to increase for $P \gg N$, the vectors encoded into the system through the connection matrix are retrieved with an efficiency almost three times better than for the spurious states.

[3]The calculation follows the same steps already depicted before, counting the "coherent" terms, that, in this case, only arise from the diagonal elements (i=j) and not from the μ=ν terms that now do not exist. The weight of the coherent part is equal to P instead of $N + P - 1$. The rest of the demonstration follows straightforward.

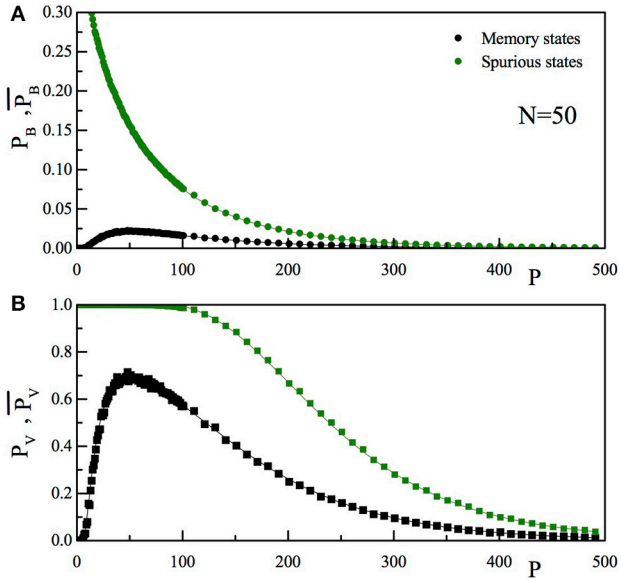

FIGURE 7 | (A) The upper panel reports for a given N-value ($N = 50$), as a function of P, the probability that, stimulating the network with a vector inside (p_B, black) or outside ($\bar{p}_B$, green) the training set, there is one bit differing between the input and the output vector. **(B)** The lower panel reports the probability that, stimulating the network with a vector inside (p_V, black) or outside ($\bar{p}_V$, green) the training set, the vector obtained after one dynamical step is not the stimulating vector. The full lines are the theoretical prediction, the full dots are the results of the numerical simulation.

4. DISCUSSION

In this work we have developed a simple theoretical approach to investigate the computational properties and the storage capacity of feed-forward networks with self-connections. We have worked out an exact expression which gives the probability p_B of having a wrong bit in the recovery of a memory element from a Hebbian N-node neural network, where P memory elements are stored. In disagreement with previous studies we have investigated the case in which the diagonal elements were not forced to vanish. Studying the storage capacity, and deriving the related probability p_V and number N_V of having a wrongly recovered memory element, we discovered that besides the well know $P \ll N$ region, there is another region, at $P \gg N$, where the recovery is highly effective. When $P \gg N$, the efficiency of recall for a large number of encoded vectors in the J matrix is related to the presence of non-zero diagonal elements of the matrix. Basically, the higher storage performance of the network depends on the number of "coherent" terms (the signal) in the quantity A_i^μ (see Section 3.1) with respect to the "incoherent" ones (the noise). The larger the ratio between coherent to incoherent terms, the lower the probability of a wrong recovery. The number of coherent terms is $(N + P - 1)$ in the case of autapses, it is $(N - 1)$ in the case of no autapses. Indeed, the P terms disappear if the diagonal is forced to be zero as in the standard Hopfield model. It is clear that, apart from a transient regime at $P \sim N$, increasing $P \gg N$ strongly reinforces the signal-to-noise ratio and induces a much larger storage capacity. In addition to the vectors encoded into

the system, other unwanted memories also appear in the network. These are the spurious states, fixed points which do not belong to the training set. The presence of spurious states is not a feature specific to the present model, it is a typical characteristic of the standard Hopfield network and its successive improvements. Indeed, as shown by Tanaka and Edwards (1980), a random $N \times N$ matrix has $2^{\gamma N}$ fixed points ($\gamma \approx 2$). As an example, if $N = 100$, the number of fixed points is about one million. A Hebbian 100×100 matrix storing $P = 1000$ patterns, besides the "good" P fixed points have also an overwhelming number of spurious fixed points (or "false memories"). The interest of our approach does not rely in "how many" spurious (i.e., not belonging to the training set) states are present but rather in how the recognition of a vector belonging to the training set is as a "good" one. Obviously, the argument of Tanaka-Edwards applies only to random matrices. The Hebbian form, with or without autapses, is not fully random (there exists correlation among the matrix elements), but we expect a number of fixed points similar to that of a random matrix. It would be interesting to determine such a number, but this is beyond the scope of the present paper. In spite of the overwhelming majority of spurious fixed points, the network—even at very large P-values, maintains the capacity of discriminate between "good" state (belonging to the training set) and "wrong" ones (not belonging to the training set). More specifically, looking at the one-step dynamical evolution and comparing the input vector with the output one, we have posed to the network the question: "is the input vector belonging to the training set"? We have demonstrated that, when the input vector actually belongs to the training set, at large P (similarly to low P) the probability of having a wrong response ("no, it does not belong to the training set) goes to zero. Furthermore, we have demonstrated that when the input vector does not belong to the training set the probability of a wrong response ("yes, it is a fixed point") is much less that in the previous case, asymptotically 2.7 time worst.

In order to identify whether or not a vector belonging to the training set was a fixed point we propose to the system a vector of the training set as input. Then we perform a one-step dynamic evolution of this input state. If after one step the output vector is equal to the input one, this is a fixed point. On the contrary, if after one step the output vector is not equal to the input one, it could be possible that further dynamical steps lead to the input vector. From this point on, as the dynamic is deterministic, the system enters a limit cycle (of length greater than one). Since it is not clear whether or not a limit cycle can be considered a "right recognition," we have excluded this possibility from the counts of the right recognition. Only fixed point are considered "good." For this reason, to determine the probability of "right recognition" one dynamical step is enough. We have also not considered the possibility that, using as input a vector not belonging to the training set, it converges to one of the training vectors. The probability of right recognition reported here is an underestimation of the network capability. A further quantity that it would be interesting to evaluate is the size of the attraction basin of a given fixed point, i.e., how many non-training vectors converge to a given training vector fixed point. The basins size would be an important measure of the network performance, their determination is however difficult to achieve analytically, and is behind the goal of the present paper.

One important finding is summarized in Equation (18). It states that for $P \gg N$, when the connection matrix is dominated by the diagonal term and is still different from the unity matrix (this is due to the great number of off-diagonal elements with zero average and RMS of the order of $1/\sqrt{P}$), the network retains its capacity of give more "good" than "wrong" answers. This property, the fact that the limit in Equation (18) is e and not "1," can be ascribed to the observation that, although the matrix **J** tends to the unit matrix for large P, the dynamics (see Equation 1) dictated by the matrix J does not tend to the dynamics dictated by the unit matrix. This finding opens the way to a much more efficient use of the artificial Hebbian neural network for information storage. In the first region, as well known since 40 years, the storage capacity is limited as the number of encoded vectors becomes of the order N. Indeed, in the high α region, the number of elements is basically unlimited [4], when the number of stored elements is taken *larger* than $\approx 4N\ln(N)$.

AUTHOR CONTRIBUTIONS

GR designed research. VF, ML, and GR performed numerical simulations, analyzed data and wrote the manuscript.

[4]The word unlimited is obviously non-physical. However, the value of P_o arising from the Tanaka-Edwards relation (Tanaka and Edwards, 1980), $P_o = \exp(\gamma N)$, with $\gamma = 0.2$. P_o, already at the N-values reported in Equation (5) is so large ($P_o(N = 1000) \approx 10^{87}$) with respect to $P(N)$ ($P(N = 1000) \approx 2\ 10^4$ to state that P_o is unlimited to any practical purpose.

REFERENCES

Abu-Mostafa, Y. S., and St. Jacques, J.-M. (1985). Information capacity of the Hopfield model. *IEEE Trans. Inf. Theory* IT-31, 461. doi: 10.1109/tit.1985.1057069

Amit, D. J. (1989). *Modelling Brain Function: The World of Attractor Neural Networks*. Cambridge: Cambridge University Press.

Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985a). Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.* 55:1530. doi: 10.1103/PhysRevLett.55.1530

Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985b). Spin-glass models of neural networks. *Phys. Rev. A* 32:1007. doi: 10.1103/physreva.32.1007

Bastolla, U., and Parisi, G. (1997). Attractors in fully asymmetric neural networks. *J. Phys. A Math. Gen.* 30:5613.

Brunel, N. (2016). Is cortical connectivity optimized for storing information? *Nat. Neurosci.* 19:749. doi: 10.1038/nn.4286

Cooper, L. N. (1973). "A possible organization of animal memory and learning," in *Proceedings of the Nobel Symposium on Collective Properties of Physical Systems,* eds B. Lundquist and S. Lundquist (New York, NY: Academic Press), 252–264.

Cooper, L. N., Liberman, F., and Oja, E. (1979). A theory for the acquisition and loss of neuron specificity in visual cortex. *Biol. Cybern.* 33:9. doi: 10.1007/BF00337414

Corless, R. M., Gonnet, G. H., Hare, D. E. G., Jeffrey, D. J., and Knuth, D. E. (1996). On the LambertW function. *Adv. Comput. Math.* 5:329. doi: 10.1007/BF02124750

Derrida, B. (1989). Distribution of the activities in a diluted neural network. *J. Phys. A Math. Gen.* 22:2069. doi: 10.1088/0305-4470/22/12/012

Eccles, J. G. (1953). *The Neurophysiological Basis of Mind*. Oxford: Clarendon.

Gutfreundt, H., Regert, J. D., and Young, A. P. (1988). The nature of attractors in an asymmetric spin glass with deterministic dynamics. *J. Phys. A Math. Gen.* 21:2775. doi: 10.1088/0305-4470/21/12/020

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall. Available online at: https://lecteurepub-a507a.firebaseapp.com/3RO4lXk7OXbEm12/Neural%20Networks%20A%20Comprehensive%20Foundation%202nd%20Edition%20Ebooks%20Gratuit.pdf

Hebb, D. O. (1949). *The Organization of Behavior*. New York, NY: Wiley.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. U.S.A.* 79:2554. doi: 10.1073/pnas.79.8.2554

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci. U.S.A.* 81:3088. doi: 10.1073/pnas.81.10.3088

Hopfield, J. J., Feinstein, D. I., and Palmer, R. G. (1983). 'Unlearning' has a stabilizing effect in collective memories *Nature* 304, 158.

Mc Eliece, R. J., Posner, E. C., Rodemich, E. R., and Venkatesh, S. S. (1987). The capacity of the hopfield associative memory. *IEEE Trans. Inf. Theory* IT-33, 461.

Olver, F. W. J., Lozier, D. W., Boisvert, R. F. and Clark, C. W. (eds.). (2010). *Handbook of Mathematical Functions*. Cambridge: Cambridge University Press.

Rojas, R. (1996). *Neural Networks*. Berlin: Springer-Verlag.

Sollacher, R., and Gao, H. (2009). Towards real-world applications of online learning spiral recurrent neural networks. *J. Intell. Learn. Syst. Appl.* 1:1. doi: 10.4236/jilsa.2009.11001

Sompolinsky, H., Crisanti, A., and Sommers, H. (1988). Chaos in random neural networks. *Phys. Rev. Lett.* 61:259. doi: 10.1103/PhysRevLett.61.259

Tanaka, F., and Edwards, S. F. (1980). Analytic theory of the ground state properties of a spin glass. I. Ising spin glass. *J. Physi. F Met. Phys.* 10:2769. doi: 10.1088/0305-4608/10/12/017

Wainrib, G., and Touboul, J. (2013). Topological and dynamical complexity of random neural networks. *Phys. Rev. Lett.* 118:101259. doi: 10.1103/physrevlett.110.118101

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.