Storage capacity of a constructive learning algorithm

Storage capacity of a constructive learning algorithm

Arnaud Buhot and Mirta B Gordon†

SPSMS/DRFMC/CEA Grenoble, 17 av. des Martyrs, 38054 Grenoble Cedex 9, France

Abstract. Upper and lower bounds for the typical storage capacity of a constructive algorithm, the tilinglike learning algorithm for the parity machine (Biehl M and Opper M 1991 *Phys. Rev. A* **44** 6888), are determined in the asymptotic limit of large training set sizes. The properties of a perceptron with threshold, learning a training set of patterns having a biased distribution of targets, needed as an intermediate step in the capacity calculation, are determined analytically. The lower bound for the capacity, determined with a cavity method, is proportional to the number of hidden units. The upper bound, obtained with the hypothesis of replica symmetry, is close to the one predicted by Mitchinson and Durbin (1989 *Biol. Cybern.* **60** 345).

1. Introduction

In this paper, we consider the problem of learning binary classification tasks from examples with neural networks. The network's architecture and the neurons' weights are determined based on a training set of examples or patterns $\mathcal{L}_\alpha$, composed of $P = \alpha N$ input vectors $\{x^\mu\}_{\mu=1,\ldots,P}$ in N-dimensional space and their corresponding classes $\tau^\mu = \pm 1$. The latter are the targets to be learned. Hereafter, we call $\alpha \equiv P/N$ the *size* of the training set. One interesting property that characterizes a neural network is its storage capacity, which is the size α_c of the largest training set with arbitrary targets the network is able to learn (with probability 1). The perceptron, a single neuron connected to its inputs through N weights, performs linear separations and has a storage capacity $\alpha_c = 2$ [1–4]. It is possible to increase the storage capacity of neural networks by considering more complicated architectures, like those with one hidden layer of k units. Such monolayer perceptrons map each input vector x to a binary k-dimensional internal representation determined by the outputs of k perceptrons, which in this context are also called hidden units. The overall network's output to an input pattern is a Boolean function of the corresponding internal representation. This function may be learned by an output perceptron, but then the internal representations of the training set must be linearly separable. In order to get rid of this constraint, networks implementing particular functions of the hidden states have been investigated. Among these, the committee machine, whose output is the class of the majority of the hidden units, and the parity machine, whose output is the product of the k components of the internal representation, have deserved particular attention [5].

Learning consists of adapting the number of hidden perceptrons and their weights in order that the outputs of the network to the training examples match the corresponding targets. The main problem is that the internal representations are unknown. Besides the CHIR algorithm [6], that determines the internal representations through a random process involving

† also at Centre National de la Recherche Scientifique.

learning faithful sets of internal representations with k fixed, most learning algorithms build the internal representations through a deterministic incremental procedure that determines k by construction. In the latter case, the hidden perceptrons are trained one after the other with targets that differ from one algorithm to another, until the correct classification is achieved. The first incremental procedure was proposed by Gallant [7]. Many other authors developed further this idea, like Mézard and Nadal with the tiling algorithm [8], Ruján and Marchand with the sequential learning algorithm [9] and Biehl and Opper with the tilinglike learning algorithm (TLA) [10]. Other variations have been proposed [11, 12]. It has been argued that these incremental procedures may require a number of hidden units much larger than the number actually needed by a network making use of its full storage capacity. In the following we thus distinguish the algorithm's capacity, defined as the size of the largest training set (with arbitrary targets) learnable with the algorithm, from the capacity of the network with the same architecture. Clearly the former cannot be larger than the latter. An upper bound for the storage capacity of the parity machine with k hidden perceptrons has been obtained by Mitchinson and Durbin [13] through a geometric approach: $\alpha_c(k) \leqslant k \ln k / \ln 2$. Recent replica calculation results, obtained in the limit of a large number of hidden perceptrons ($k \to +\infty$) [14], strongly suggest that this upper bound may effectively be reached. However, the learning problem remains: is there a learning algorithm whose capacity saturates this bound? This question was addressed recently in [15] within the same statistical mechanics framework as this work. In spite of a thorough analysis, no clear-cut conclusion could be drawn in the asymptotic regime of large k, because of a lack of precision in the numerical integration of the corresponding equations.

In this paper, we determine analytically the storage capacity of a parity machine built with the TLA. Our results present strong evidence showing that the storage capacity of the obtained network is close to the upper bound, at least within the replica-symmetry (RS) approximation. The paper is organized as follows: in section 2, we describe the TLA. The conditions necessary for the TLA to converge impose strong constraints on the cost function used to train the hidden perceptrons. These are discussed in section 3. Despite intensive research in this field, no analytic results on the learning properties of the perceptron with threshold, in the asymptotic limit $\alpha \to +\infty$ needed here, exist. These are deduced in section 4 for the Gardner cost function with vanishing and finite margin, within the RS approximation. As this approximation is known to provide only a lower bound to the perceptron's actual training error [16, 17], we also determined an upper bound through a generalization of the Kuhn–Tucker (KT) cavity method proposed by Gerl and Krey [18]. The general expression for the number of hidden perceptrons generated by the TLA in the limit $\alpha \to +\infty$ is deduced in section 5. Our main result is that the number of hidden units needed by the TLA to converge grows proportionally to $\alpha/(\ln \alpha)^\nu$ in the large α limit, where $\nu = 1$ in the RS approximation and $\nu = 0$ within the KT cavity method, provided that the hidden perceptrons learn through the minimization of their training errors. Our results are discussed and compared both with the Mitchinson and Durbin bound [13] and with the numerical results obtained by West and Saad [15]. The general conclusion is left to section 6.

2. The tilinglike learning algorithm

In the following, we describe the TLA, considered here because of its simplicity. The TLA needs hidden perceptrons with a threshold to generate the parity machine. The classification performed by a perceptron is a linear separation defined by a hyperplane in the N-dimensional input space, of normal vector $\boldsymbol{J}$ ($\boldsymbol{J} \cdot \boldsymbol{J} = 1$) and distance to the origin θ. The N components

of $\boldsymbol{J}$ are the perceptron's weights and θ is its threshold. An example $\boldsymbol{x}$ is classified as follows:

$$\sigma \equiv \text{sign}(\boldsymbol{J} \cdot \boldsymbol{x} - \theta). \tag{1}$$

As already pointed out in [15] the threshold is useful in the case of unbalanced training sets, containing more examples of one class than of the other. As we will see in the following, this is the case for the successive perceptrons included by the TLA.

In the first learning step of the algorithm, the parameters $\boldsymbol{J}_1$ and θ_1 of a perceptron are adapted in order to obtain the lowest possible number of training errors. This is usually done through the minimization of a cost function:

$$E(\boldsymbol{J}_1, \theta_1; \mathcal{L}_\alpha) = \sum_{\mu=1}^{P} V(\lambda_1^\mu) \tag{2}$$

where the potential V is a function of λ_1^μ, the stability of the example μ:

$$\lambda_1^\mu \equiv \tau^\mu (\boldsymbol{J}_1 \cdot \boldsymbol{x}^\mu - \theta_1). \tag{3}$$

The stability is positive if and only if the example is correctly classified. Its absolute value is the distance of the example to the separating hyperplane.

In principle, there is some freedom in the choice of the potential $V(\lambda)$. As it has to penalize training errors, it has to be a decreasing function of λ. Considering as cost function the number of training errors corresponds to the particular choice $V(\lambda) = \Theta(-\lambda)$, where $\Theta(x)$ is the Heaviside function. Other potentials, that do not minimize the number of training errors but possess interesting learning or algorithmic properties may be chosen. Examples are $V(\lambda) = (\kappa - \lambda)^n \Theta(\kappa - \lambda)$ where $\kappa \geqslant 0$ is a fixed positive margin chosen *a priori*. The case $n = 0$ corresponds to the so-called Gardner potential [3, 4] which reduces to the error counting function for $\kappa = 0$. The potential defined by $n = 1$ corresponds to the perceptron learning algorithm [19–21] and $n = 2$ to the AdaTron [21, 22].

After learning, the training error of the first perceptron is:

$$\varepsilon_t^1(\boldsymbol{J}_1^*, \theta_1^*; \mathcal{L}_\alpha) = \frac{1}{P} \sum_{\mu=1}^{P} \Theta(-\tau^\mu \sigma_1^\mu) \tag{4}$$

where σ_1^μ, the class given by the perceptron to the example $\boldsymbol{x}^\mu$, depends through equation (1) on the parameters $\boldsymbol{J}_1^*$ and θ_1^* that minimize the cost function.

If the training error is zero, the learning procedure stops. Then, the class associated to the patterns by the parity machine is just the class given by the first perceptron. Otherwise, another perceptron is included and trained with the aim of separating the correctly learned examples from the wrongly learned ones. The corresponding training set $\mathcal{L}_\alpha^2 = \{\boldsymbol{x}^\mu, \tau_2^\mu\}_{\mu=1,\dots,P}$ contains the same input examples as $\mathcal{L}_\alpha$ with new targets τ_2^μ defined as follows: $\tau_2^\mu = +1$ if the example $\boldsymbol{x}^\mu$ is correctly classified by the previous perceptron and $\tau_2^\mu = -1$ if not. These targets may be expressed as $\tau_2^\mu = \sigma_1^\mu \tau^\mu$. Notice that a fraction $1 - \varepsilon_t^1$ of patterns have targets $+1$, and a fraction ε_t^1 have targets -1. Since we expect the training error ε_t^1 to be smaller than $\frac{1}{2}$, the probability of targets -1 is smaller than that of targets $+1$. The successive perceptrons need a threshold to learn such biased training sets. Otherwise, the tilinglike construction cannot converge.

The parameters $\boldsymbol{J}_2^*$ and θ_2^* of the second perceptron are learned with the training set $\mathcal{L}_\alpha^2$, minimizing the same cost function as the first one. The same procedure, in which the perceptron $i + 1$ learns the training set $\mathcal{L}_\alpha^{i+1} = \{\boldsymbol{x}^\mu, \tau_{i+1}^\mu = \tau_i^\mu \sigma_i^\mu\}_{\mu=1,\dots,P}$, has to be iterated until $\varepsilon_t^k = 0$. Then, the product σ^μ of the classes $\{\sigma_i^\mu\}_{i=1,\dots,k}$ given by the hidden perceptrons to an example $\boldsymbol{x}^\mu$ corresponds to the target τ^μ [10, 12], as $\sigma^\mu \equiv \sigma_1^\mu \cdots \sigma_k^\mu = \sigma_1^\mu \cdots \sigma_{k-2}^\mu (\sigma_{k-1}^\mu)^2 \tau_{k-1}^\mu = \sigma_1^\mu \cdots \sigma_{k-2}^\mu \tau_{k-1}^\mu = \cdots = \tau^\mu$. Thus, the TLA constructs a parity machine with k hidden units.

3. Convergence conditions

It has been shown that if the examples are binary [10], or real-valued vectors in general position [23], there is a solution that satisfies the TLA construction with the property that $P\varepsilon_t^i$ is a succession of decreasing integer numbers. Thus, a finite $k \leqslant P$ exists for which $\varepsilon_t^k = 0$.

In the following, we are interested in the *typical* number k of hidden perceptrons necessary for the TLA to learn a training set of size α. This is obtained in the thermodynamic limit where N and P diverge keeping $\alpha = P/N$ constant. In this limit, k is expected to be independent of the particular set of training patterns, and to depend only on α. However, as $P \to +\infty$, it is not possible to argue that $P\varepsilon_t^i$ is a succession of strictly decreasing numbers in order to guarantee the convergence of the TLA in a finite number of steps (i.e. of hidden units). In particular, the solution in which a single example is correctly learned at each step, used by the convergence proofs [10, 23] at finite N, leads to $k \to +\infty$. In order to obtain a *finite* number $k(\alpha)$ in the thermodynamic limit, each perceptron has to learn at least a number of examples of the order of N. This imposes some general conditions on the learning algorithm used to train the perceptrons.

It is worth pointing out that the conditions for convergence with finite k in the thermodynamic limit do not guarantee the convergence for *all* the possible training sets of size α. This is due to the probabilistic nature of the statistical physics results, which predict the average behaviour. The results may not be correct for subsets of zero measure in the space of training sets, and in particular for the worst case.

As described before, the training set $\mathcal{L}_\alpha^i$ used to train the perceptron i contains a fraction $1 - \varepsilon_t^{i-1}$ of patterns with targets $+1$, and a fraction ε_t^{i-1} of patterns with targets -1. These targets are slightly correlated, as they are determined by the training errors of the preceding perceptron. However, it has been shown that these correlations are weak [15]. We neglect them in the limit $\alpha \to +\infty$ considered in the following. Thus, we consider that the targets to be learned by the successive perceptrons are i.i.d. random variables, and have a probability $1 - \varepsilon_t^{i-1}$ to be $+1$ and ε_t^{i-1} to be -1. As this neglects the constraints imposed by the correlations on the minimization of the training error, we expect that the assumption of uncorrelated targets underestimate the perceptrons' training errors. It follows that our estimation of the number $k(\alpha)$ of perceptrons necessary to construct the parity machine is a lower bound to the actual value.

Consider a perceptron learning a training set of size α with targets given by the following biased probability distribution:

$$P(\tau) = (1 - \varepsilon)\delta(\tau - 1) + \varepsilon\delta(\tau + 1). \tag{5}$$

If $\mathcal{E}_t(\alpha, \varepsilon)$ is the perceptron's training error, i.e. the fraction of wrongly learned examples, there is a simple relationship between the training errors ε_t^{i-1} and ε_t^i of two successive hidden perceptrons:

$$\varepsilon_t^i = \mathcal{E}_t(\alpha, \varepsilon_t^{i-1}) \tag{6}$$

since the bias in the probability of the targets of perceptron i is due to the training error of the preceding unit.

The successive training errors ε_t^i must decrease monotonically with i and eventually vanish for a finite k. Otherwise the TLA does not converge. Taking equation (6) into account, this imposes that

$$\mathcal{E}_t(\alpha, \varepsilon) < \varepsilon. \tag{7}$$

Condition (7) restricts the possible potentials in the cost function (2). For example, in the following section we show that the perceptron and the AdaTron potentials [19–22] do not satisfy the condition (7) for all α when $\varepsilon < \frac{1}{2}$.

The stopping condition of the TLA imposes that there is a finite value of k such that

$$\varepsilon_t^k = \mathcal{E}_t(\alpha, \varepsilon_t^{k-1}) = 0. \tag{8}$$

This in turn imposes that for all α, there always exists $\varepsilon_0(\alpha) \neq 0$ such that $\mathcal{E}_t(\alpha, \varepsilon_0(\alpha)) = 0$. Thus, the stopping condition (8) imposes that the inverse function $\alpha_0(\varepsilon)$ diverges as $\varepsilon \to 0$. In fact, $\alpha_0(\varepsilon)$ is the storage capacity of a perceptron learning targets drawn with the biased probability (5) (in the literature, the bias is usually defined as $1 - 2\varepsilon$). Actually, the divergence of $\alpha_0(\varepsilon)$ occurs whenever the potential $V(\lambda)$ vanishes for $\lambda > 0$ and is strictly positive for $\lambda < 0$. This is the case for the Gardner potential with $\kappa = 0$, for which $\alpha_0(\varepsilon) \sim -(\varepsilon \ln \varepsilon)^{-1}$ [2–4]. However, even if the perceptron has been extensively studied, very few results exist for the case of training sets with biased distributions of targets [3,10,24]. In particular, the asymptotic behaviour of the learning curves $\mathcal{E}_t(\alpha, \varepsilon)$ as a function of α is unknown. These are deduced in the next section. The reader not interested in these intermediate calculations may skip them and go straight to section 5. Only the results displayed by equations (24), (30), (45) and (48) are used to determine the asymptotic behaviour of the TLA.

4. Perceptron's training error for biased target distributions

In order to learn training sets with biased distributions of targets, the perceptron must have a threshold, as the separating hyperplanes that minimize the training error do not contain the origin. Here we present new analytic results, mainly in the asymptotic regime $\alpha \to +\infty$, for the Gardner cost function defined by the potential

$$V(\lambda) = \Theta(\kappa - \lambda). \tag{9}$$

For $\kappa = 0$, the corresponding cost function is the number of training errors. For $\kappa > 0$, the cost function is the number of examples with stability (3) smaller than κ.

The section is divided into two parts. In the first one we derive results within the RS approximation, which is known to underestimate the training error. In the second part we obtain upper bounds for the training error, using a cavity method.

4.1. Replica calculation

We briefly recall the main steps of the replica calculation, that follows the same lines as [10,24]. As we are interested in the properties of the minimum of the cost function, a temperature $T \equiv 1/\beta$ is introduced and the cost function is considered as an energy. The corresponding partition function is

$$Z(\beta, \mathcal{L}_\alpha(\varepsilon)) = \int d\theta \, P(\theta) \int dJ \, P(J) \exp(-\beta E(J, \theta; \mathcal{L}_\alpha(\varepsilon))) \tag{10}$$

where the components of J are the weights, and θ is the perceptron's threshold. $\mathcal{L}_\alpha(\varepsilon)$ is a training set of size α. The input vectors x^μ are drawn from a Gaussian distribution with zero mean and unit variance in all the directions. The targets have the biased distribution (5).

Following Gardner's approach, the patterns of the training set are considered as frozen disordered variables. The replica trick allows to calculate the mean free energy in the thermodynamic limit ($N \to +\infty$, $P \to +\infty$ and α constant) averaged over all possible training sets, as follows:

$$f(\alpha, \varepsilon) = \lim_{\beta \to +\infty} \lim_{\substack{N \to +\infty \\ P \to +\infty \\ \alpha = P/N}} \lim_{n \to 0} -\frac{1}{\beta n N} \ln \overline{Z^n(\beta, \mathcal{L}_\alpha(\varepsilon))} \tag{11}$$

where the bar stands for the mean over the training sets with same size α. Thus, the free energy is obtained through the averaging of a partition function of n replicas of the original system. Hereafter we assume RS, i.e. that the replicas are equivalent under permutation. However, it is well known that RS breaks down when the training error is finite [25]. Calculations including one step of RS breaking have shown that the training error obtained within the RS approximation is a lower bound for the actual one [16, 17].

Assuming that the weights have a uniform prior probability over the surface of the N-dimensional sphere of unitary radius, and the threshold a uniform distribution over the real axis between $-\sqrt{N}$ and $+\sqrt{N}$, the free energy within the RS approximation is

$$f(\alpha, \varepsilon) = \max_c \min_\theta g(\alpha, \varepsilon, c, \theta) \tag{12}$$

where the function g is

$$g(\alpha, \varepsilon, c, \theta) = -\frac{1}{2c} + \alpha(1 - \varepsilon) \int W(\lambda(y, c), y, c) \exp\left(-\frac{(y + \theta)^2}{2}\right) \frac{\mathrm{d}y}{\sqrt{2\pi}}$$

$$+ \alpha\varepsilon \int W(\lambda(y, c), y, c) \exp\left(-\frac{(y - \theta)^2}{2}\right) \frac{\mathrm{d}y}{\sqrt{2\pi}} \tag{13}$$

with $\lambda(y, c)$ the function that minimizes: $W(\lambda, y, c) \equiv V(\lambda) + (\lambda - y)^2/2c$. c is the usual order parameter in replica calculations ($c = \lim_{\beta \to +\infty} \beta(1 - \boldsymbol{J}_a \cdot \boldsymbol{J}_b)$ with $\boldsymbol{J}_a$ and $\boldsymbol{J}_b$ the directions corresponding to two different replicas). The parameters c and θ are solutions of the following extremum conditions

$$\frac{\partial g}{\partial c} = \frac{\partial g}{\partial \theta} = 0. \tag{14}$$

The training error $\mathcal{E}_t(\alpha, \varepsilon)$ may be easily deduced by integration of the distribution of stabilities over the negative values [24], yielding

$$\mathcal{E}_t(\alpha, \varepsilon) = (1 - \varepsilon) \int \Theta(-\lambda(y, c)) \exp\left(-\frac{(y + \theta)^2}{2}\right) \frac{\mathrm{d}y}{\sqrt{2\pi}}$$

$$+ \varepsilon \int \Theta(-\lambda(y, c)) \exp\left(-\frac{(y - \theta)^2}{2}\right) \frac{\mathrm{d}y}{\sqrt{2\pi}}. \tag{15}$$

Equations (10)–(15) are valid for any potential $V(\lambda)$ in (2). In the following, we concentrate specifically on the Gardner potential (9). The function $\lambda(y, c)$ that minimizes $W(\lambda, y, c)$ for a given κ is

$$\lambda(y, c) = \begin{cases} y & \text{for} \quad y < \kappa - \sqrt{2c} \\ \kappa & \text{for} \quad \kappa - \sqrt{2c} < y < \kappa \\ y & \text{for} \quad \kappa < y. \end{cases} \tag{16}$$

Introducing (16) into (13), we deduce $g(\alpha, \varepsilon, c, \theta)$. The conditions (14) allow us to determine the equations for c and θ:

$$\frac{1}{\alpha} = (1 - \varepsilon) \int_{\kappa - \sqrt{2c} + \theta}^{\kappa + \theta} (\kappa + \theta - y)^2 \mathrm{D}y + \varepsilon \int_{\kappa - \sqrt{2c} - \theta}^{\kappa - \theta} (\kappa - \theta - y)^2 \mathrm{D}y \tag{17}$$

$$0 = (1 - \varepsilon) \int_{\kappa - \sqrt{2c} + \theta}^{\kappa + \theta} (\kappa + \theta - y) \mathrm{D}y - \varepsilon \int_{\kappa - \sqrt{2c} - \theta}^{\kappa - \theta} (\kappa - \theta - y) \mathrm{D}y \tag{18}$$

where $\mathrm{D}y = \exp(-y^2/2) \, \mathrm{d}y/\sqrt{2\pi}$. The distribution of stabilities of the training patterns is $\rho(\lambda) = (1 - \varepsilon)\rho_+(\lambda) + \varepsilon\rho_-(\lambda)$ with

$$\rho_\pm(\lambda) = \delta(\lambda - \kappa) \int_{\kappa - \sqrt{2c} \pm \theta}^{\kappa \pm \theta} \mathrm{D}y + \{\Theta(\kappa - \sqrt{2c} - \lambda) + \Theta(\lambda - \kappa)\} \exp\left(-\frac{(\lambda \pm \theta)^2}{2}\right) \frac{1}{\sqrt{2\pi}}. \tag{19}$$

$\rho(\lambda)$ presents a two-band structure with a gap between $\lambda_- = \kappa - \sqrt{2c}$ and $\lambda_+ = \kappa$. Notice that only if $\lambda_- < 0$ the lower band corresponds to wrongly classified patterns. If $\kappa > 0$, then λ_- may become positive for sufficiently small values of c. In that case, the training error is only a fraction of the patterns lying in the lower band. Taking this into account, the training error $\mathcal{E}_t(\alpha, \varepsilon)$ is

$$\mathcal{E}_t(\alpha, \varepsilon) = (1 - \varepsilon) \int_{\max[-\theta, \sqrt{2c}-\kappa-\theta]}^{+\infty} \mathrm{D}y + \varepsilon \int_{\max[\theta, \sqrt{2c}-\kappa+\theta]}^{+\infty} \mathrm{D}y. \tag{20}$$

We derive separately the asymptotic properties for $\kappa = 0$ and for $\kappa \neq 0$, for reasons that will become clear in the following.

We consider first the case $\kappa = 0$. The band of positive stabilities starts at $\lambda_+ = 0$ so that the gap, of width $\sqrt{2c}$, lies strictly in the region of negative stabilities. As we expect that the gap vanishes for $\alpha \to +\infty$, we look for solutions of the extremum equations with $c \to 0$ and $|\theta| \to +\infty$ (notice that θ is negative for $\varepsilon < \frac{1}{2}$) with the product $a = \theta\sqrt{2c}$ finite. Introducing these assumptions into (18), we determine ε as a function of a:

$$\varepsilon = \frac{e^a(1-a) - 1}{2(\cosh a - a \sinh a - 1)}. \tag{21}$$

The relation between α, θ and a follows from (17) and (21):

$$\frac{1}{\alpha} = \frac{\exp(-\theta^2/2)}{\theta^3 \sqrt{2\pi}} \left\{ \frac{a^2(\sinh a - a)}{\cosh a - a \sinh a - 1} \right\}. \tag{22}$$

a and θ are increasing functions of ε as expected. For a symmetric distribution of targets ($\varepsilon = \frac{1}{2}$) then $a = 0$ corresponding to a vanishing threshold. Conversely, if all the targets are $+1$ ($\varepsilon = 0$), the threshold diverges to $-\infty$. For finite $\varepsilon < \frac{1}{2}$, the absolute value of the threshold is an increasing function of α. From equation (22) we obtain the development $\theta^2 = 2\ln\alpha + O(\ln\ln\alpha)$. Notice that neglecting $\ln\ln\alpha$ with respect to $\ln\alpha$ is an approximation only valid for large enough α ($\alpha > 10^{10}$). As was already pointed out in [24], this behaviour cannot be deduced by solving the equations (17) and (18) numerically.

The training error $\mathcal{E}_t(\alpha, \varepsilon)$ (20) with $\kappa = 0$ in the limit $\alpha \to +\infty$ is then

$$\mathcal{E}_t(\alpha, \varepsilon) \simeq \varepsilon - \frac{\exp(-\theta^2/2)}{\theta\sqrt{2\pi}} \left\{ \frac{\sinh a - a}{\cosh a - a \sinh a - 1} \right\}. \tag{23}$$

Using equations (22) and (23), we deduce that

$$\mathcal{E}_t(\alpha, \varepsilon) \simeq \varepsilon - \frac{\theta^2}{\alpha a^2(\varepsilon)} \simeq \varepsilon - \frac{\ln\alpha}{\alpha} \frac{2}{a^2(\varepsilon)} \tag{24}$$

where $a(\varepsilon)$ is the inverse function of $\varepsilon(a)$ given by (21).

Consider now the Gardner potential with finite κ. Although a solution of equations (17) and (18) under the assumption that $c \to 0$ with finite θ in the limit $\alpha \to +\infty$ exists, it does *not* correspond to the correct extremum of g (13). It is, however, worth examining. The corresponding value of θ as a function of ε and κ follows from (18), and the relation between α, θ, κ and c from (17). We find that

$$\varepsilon = \frac{1}{1 + \exp(2\kappa\theta)} \tag{25}$$

$$\frac{1}{\alpha} = \frac{2\varepsilon(2c)^{3/2}}{3\sqrt{2\pi}} \exp\left(-\frac{(\kappa+\theta)^2}{2}\right). \tag{26}$$

As $\sqrt{2c} < \kappa$, the training error given by equation (20) is

$$\mathcal{E}_t(\alpha, \varepsilon) = \varepsilon + (1 - 2\varepsilon) \int_{-\theta}^{+\infty} \mathrm{D}y \tag{27}$$

and is larger than ε for any finite θ. Note that this (incorrect) solution does not satisfy the condition (7) necessary for the TLA to converge.

In fact, the correct training error corresponds to a solution with finite gap ($\sqrt{2c} \to 2\kappa$) and a diverging threshold ($\theta \to -\infty$) in the large α limit. Defining $\delta \equiv 2\kappa - \sqrt{2c}$, and keeping only the leading terms, equations (17), (18) and (20) for $\kappa > 0$ give

$$\frac{\varepsilon}{1 - \varepsilon} \simeq -\frac{\exp(\delta(\theta + \kappa))}{2\kappa(\theta + \kappa)} \tag{28}$$

$$\frac{1}{\alpha} \simeq \frac{2\kappa(1 - \varepsilon)}{(\theta + \kappa)^2 \sqrt{2\pi}} \exp\left(-\frac{(\theta + \kappa)^2}{2}\right) \tag{29}$$

$$\mathcal{E}_t(\alpha, \varepsilon) \simeq \varepsilon - \frac{1}{(2\kappa)^2} \frac{1}{\alpha}. \tag{30}$$

The neglected terms are $O(\exp(-2\kappa\sqrt{2\ln\alpha} + \ln\ln\alpha))$, which are only negligible if κ is finite. The prefactor $1/(2\kappa)^2$ in (30), which diverges when $\kappa \to 0$, reflects the existence of the different behaviours for vanishing and for finite κ.

This second solution only exists for bounded potentials. The perceptron and the AdaTron potentials diverge for $\lambda \to -\infty$, and the corresponding training errors become larger than ε in the large α limit. Thus, if these learning algorithms were used to train the hidden perceptrons, the TLA would not converge.

Although the case of unbiased targets (i.e. $\varepsilon = \frac{1}{2}$) is not essential for our study, we include here the corresponding analytic results for the sake of completeness. In this case, the free energy g (13) is invariant with respect to the threshold symmetry $\theta \leftrightarrow -\theta$. Thus, $\theta = 0$ is a trivial extremum of g. However, as already discussed by West and Saad in [24], two new solutions breaking the threshold symmetry appear above a given training set size α_θ. The analytical expression of α_θ may be deduced under the assumption that the two different solutions appear continuously at α_θ, as in usual second-order phase transitions, through a series expansion of the free energy in powers of θ:

$$g(\alpha, \varepsilon, c, \theta) = g(\alpha, \varepsilon, c, 0) + \frac{\theta^2}{2} \left.\frac{\partial^2 g}{\partial\theta^2}\right|_{\theta=0} + \frac{\theta^4}{24} \left.\frac{\partial^4 g}{\partial\theta^4}\right|_{\theta=0}. \tag{31}$$

Due to the symmetry, the odd derivatives with respect to θ vanish. The condition

$$\frac{\partial^2 g}{\partial\theta^2} = 0 = \int_{\kappa-\sqrt{2c}}^{\kappa} y(\kappa - y)\mathrm{D}y \tag{32}$$

defines $\sqrt{2c}$ at the transition. The size α_θ satisfies

$$\alpha_\theta = \left(\int_{\kappa-\sqrt{2c}}^{\kappa} (\kappa - y)^2 \mathrm{D}y\right)^{-1} \tag{33}$$

and the two new solutions that appear for $\alpha > \alpha_\theta$ correspond to a threshold $\theta_\pm \sim \pm\sqrt{\alpha - \alpha_\theta}$. Note that the usual stability criterion for second-order phase transitions, $\partial^4 g/\partial\theta^4 > 0$, cannot be directly applied here because we have two order parameters. Taking into account the leading corrections to c, proportional to θ^2, it is straightforward to verify that the solutions with finite threshold are stable.

4.2. Kuhn–Tucker cavity method

In order to circumvent the RS approximation, we determine the training error $\mathcal{E}_t(\alpha, \varepsilon)$ using the KT cavity method proposed by Gerl and Krey [18], that we generalize here to the case of a perceptron with a threshold learning a training set with a biased probability of targets

given by (5). Contrary to the RS solution, this cavity method has been shown to overestimate the training error [18]. Consequently, the results allow us to deduce an upper bound for the number of perceptrons needed by the tilinglike procedure to converge.

With the KT cavity method, the properties of the perceptron are determined by analysing self-consistently its response to the introduction of a new pattern into the training set. It is particularly adapted to study the properties of the Gardner potential (9) because it is based on the fact that the weights minimizing the corresponding cost function are a (conveniently normalized) linear combination of the patterns with stability κ, which are called *support vectors*.

Let us assume that the perceptron has learned the training set and that the value of the cost function is E. This is the number of examples with stability smaller than the margin κ. The support vectors belong to the subset of $\alpha N - E$ remaining examples that do not contribute to the cost. The perceptron's weights may be expressed as follows:

$$J = \frac{1}{N} \sum_{\mu \in \{\alpha N - E\}} \tau^\mu a^\mu x^\mu \tag{34}$$

with $a^\mu > 0$ for $\lambda^\mu = \kappa$, and $a^\mu = 0$ for $\lambda^\mu > \kappa$. These are the so-called KT conditions. Defining $a^\mu = 0$ for examples with $\lambda^\mu < \kappa$, the normalization of the weights imposes that

$$1 = J \cdot J = \frac{1}{N} \sum_{\mu=1}^{\alpha N} \tau^\mu a^\mu x^\mu \cdot J = \frac{1}{N} \sum_{\mu=1}^{\alpha N} a^\mu (\kappa + \tau^\mu \theta). \tag{35}$$

As usual with cavity methods, we introduce a new example x^0 with target τ^0, drawn respectively with the same probability densities as the other inputs and targets in the training set. Before any modification, as the pattern 0 is uncorrelated with the direction J and its components are assumed to have a Gaussian distribution, its projection onto J has a Gaussian probability. Therefore, the joint probability distribution of the target τ^0 and the stability $\tilde{\lambda}^0 = \tau^0 (x^0 \cdot J - \theta)$ before learning is

$$\Pi(\tilde{\lambda}^0, \tau^0) = \frac{P(\tau^0)}{\sqrt{2\pi}} \exp\left(-\frac{(\tilde{\lambda}^0 + \tau^0 \theta)^2}{2}\right) \tag{36}$$

where $P(\tau^0)$ is defined by (5). We assume a single ground state and we calculate the necessary adjustments of the weights J in order to obtain self-consistent equations for the cost function as a function of α.

If $\tilde{\lambda}^0 \geqslant \kappa$, no learning is needed, as the new example does not contribute to the cost. If $\tilde{\lambda}^0 < \kappa$, two different situations may occur. Either the distance of the new example to the hyperplane is too large and the perceptron is unable to learn it, or the example is close enough and can be learned. The natural strategy to minimize the cost function is to include the new example in the subset of support vectors only if $\kappa - \sqrt{2c} < \tilde{\lambda}^0 < \kappa$, where $\sqrt{2c}$ is a positive quantity which has to be determined self-consistently. Otherwise, the weights are not modified and the new example is left in the subset of examples contributing to the cost. We are left with the problem of determining the perturbation on the weights such that examples with $\kappa - \sqrt{2c} < \tilde{\lambda}^0 < \kappa$ become support vectors after learning. As a first step, this can be obtained by taking $a^0 = \kappa - \tilde{\lambda}^0$. However, this modifies the stabilities of the other support vectors. The coefficients $a^\mu > 0$ ($\mu \geqslant 1$) must be corrected by a small amount to compensate for this perturbation. This correction in turn modifies the stability of the new example 0, and a^0 has to be corrected. After a full summation of the contributions, Gerl and Krey [18] have shown that the correct value of a^0 is

$$a^0 = \frac{\kappa - \tilde{\lambda}^0}{1 - \alpha P(a^\mu > 0)} \tag{37}$$

where $P(a^\mu > 0)$ is the probability that $a^\mu > 0$. This probability is determined assuming that the new example is equivalent to the others:

$$P(a^\mu > 0) \equiv P(a^0 > 0) = \sum_{\tau^0=\pm 1} \int_{\kappa-\sqrt{2c}}^{\kappa} \Pi(\tilde{\lambda}^0, \tau^0)\,\mathrm{d}\tilde{\lambda}^0. \tag{38}$$

Having specified the learning procedure, we are able to determine $\sqrt{2c}$ and E self-consistently. First of all, the normalization of the weights given by equation (35), may be written as follows:

$$1 = \alpha \sum_{\tau^0=\pm 1} \int_{-\infty}^{+\infty} a^0(\kappa + \tau^0\theta)\Pi(\tilde{\lambda}^0, \tau^0)\,\mathrm{d}\tilde{\lambda}^0 \tag{39}$$

with a^0 given by (37) for $\kappa - \sqrt{2c} < \tilde{\lambda}^0 < \kappa$ and $a^0 = 0$ elsewhere. Combining equations (38) and (39), we obtain

$$1 = \alpha(1-\varepsilon)\int_{\kappa-\sqrt{2c}+\theta}^{\kappa+\theta} (1 + (\kappa + \theta)(\kappa + \theta - y))\mathrm{D}y$$

$$+\alpha\varepsilon \int_{\kappa-\sqrt{2c}-\theta}^{\kappa-\theta} (1 + (\kappa - \theta)(\kappa - \theta - y))\mathrm{D}y. \tag{40}$$

This equation, which determines $\sqrt{2c}$ for a fixed threshold θ, is slightly different from the RS result (17). The cost function E is determined assuming that it remains unchanged (to order $\sqrt{N}$) upon learning the new example. Thus, the cost per example is

$$\frac{E}{\alpha N} = \sum_{\tau^0=\pm 1} \int_{-\infty}^{\kappa-\sqrt{2c}} \Pi(\tilde{\lambda}^0, \tau^0)\,\mathrm{d}\tilde{\lambda}^0$$

$$= (1-\varepsilon)\int_{-\infty}^{\kappa-\sqrt{2c}+\theta} \mathrm{D}y + \varepsilon \int_{-\infty}^{\kappa-\sqrt{2c}-\theta} \mathrm{D}y. \tag{41}$$

Note that when $\kappa - \sqrt{2c} < 0$, $E/(\alpha N)$ (41) represents the fraction of training errors $\mathcal{E}_t(\alpha, \varepsilon)$ and is similar to (20). The threshold θ may be optimized in order to minimize the cost function:

$$\frac{\partial E}{\partial \theta} = 0. \tag{42}$$

In the following, we solve (40) and (42) in the large α limit. First of all, we consider the case $\kappa = 0$. In this case, $E/(\alpha N)$ (equation (41)) is the training error $\mathcal{E}_t$. As for the RS calculation, we may assume $\sqrt{2c} \ll |\theta|$ and $a = \theta\sqrt{2c}$ finite. We obtain the following equations:

$$\frac{\varepsilon}{1-\varepsilon} \simeq \mathrm{e}^{2a}\left(\sqrt{4a^2+1} - 2a\right) \tag{43}$$

$$\frac{1}{\alpha} \simeq a\{(1-\varepsilon)\mathrm{e}^a + \varepsilon\mathrm{e}^{-a}\}\frac{\exp(-\theta^2/2)}{\theta\sqrt{2\pi}} \tag{44}$$

$$\mathcal{E}_t(\alpha, \varepsilon) \simeq \varepsilon - \frac{F(a)}{\alpha} \simeq \varepsilon - \frac{1}{\alpha}\frac{(1 + 2a - \sqrt{4a^2+1})}{a(1 + \sqrt{4a^2+1} - 2a)}. \tag{45}$$

These results differ from those obtained with the RS calculation (equations (21), (22) and (24)).

In the case of finite margin κ, the pertinent assumptions in the large α limit are $\sqrt{2c} \to 2\kappa$ and $\theta \to -\infty$. With these, here again $E/(\alpha N)$ (41) is the training error. Defining
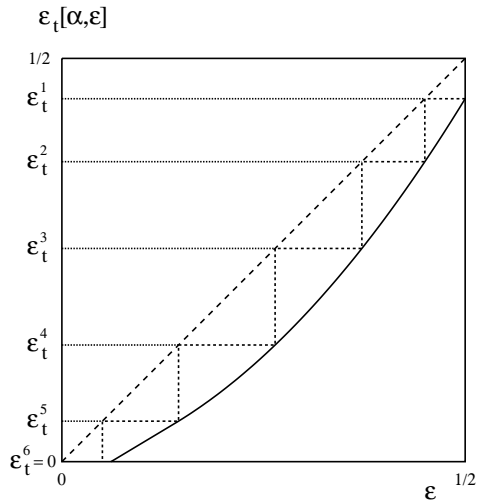
Figure 1. Evolution of the successive training errors. The full curve corresponds to the training error $\mathcal{E}_t(\alpha, \varepsilon)$ of a perceptron with biased targets. The first training error ε_t^1 is given by $\mathcal{E}_t(\alpha, \frac{1}{2})$ and the following ones by the relation $\varepsilon_t^{i+1} = \mathcal{E}_t(\alpha, \varepsilon_t^i)$. In this case, the learning algorithm converges with six perceptrons.

$\delta = 2\kappa - \sqrt{2c}$, we get

$$\frac{\varepsilon}{1-\varepsilon} \simeq \frac{4\exp(\delta(\theta+\kappa))}{(\theta+\kappa)^2} \tag{46}$$

$$\frac{1}{\alpha} \simeq \frac{8\kappa(1-\varepsilon)}{(\theta+\kappa)^2\sqrt{2\pi}} \exp\left(-\frac{(\theta+\kappa)^2}{2}\right) \tag{47}$$

$$\mathcal{E}_t(\alpha, \varepsilon) \simeq \varepsilon + \frac{1}{2\kappa\alpha(\theta+\kappa)} \simeq \varepsilon - \frac{1}{2\kappa\alpha\sqrt{2\ln\alpha}}. \tag{48}$$

It is worth pointing out that even within the KT cavity method, the training error satisfies the convergence conditions (7) and (8).

The main conclusion of this section is that the TLA converges provided that the hidden perceptrons are trained through the minimization of a cost function with a bounded potential. The Gardner potential (9) satisfies this constraint. The asymptotic behaviours of the training error in the large α limit, calculated for $\kappa = 0$ and $\kappa \neq 0$ using two different approaches are used in the following sections to characterize the storage capacity of the constructive algorithm.

5. Number of hidden perceptrons in the large α limit

We assume that the probability distribution of the targets τ^μ in the training set is symmetric, given by (5) with $\varepsilon = \frac{1}{2}$, so that the training error of the first perceptron is $\varepsilon_t^1 = \mathcal{E}_t(\alpha, \frac{1}{2})$. Considering iteratively the relationship between the training errors of two consecutive perceptrons (6) yields

$$\circ_k f_\alpha(\tfrac{1}{2}) = \underbrace{f_\alpha \circ \cdots \circ f_\alpha}_{k \text{ times}}(\tfrac{1}{2}) = 0 \tag{49}$$

where $f_\alpha(\varepsilon)$ stands for $\mathcal{E}_t(\alpha, \varepsilon)$, the symbol $\circ$ is for the composition of functions and k is the number of perceptrons necessary for convergence of the TLA algorithm.

The evolution of the training errors of the successive perceptrons is schematically represented on figure 1 for an arbitrary function $\mathcal{E}_t(\alpha, \varepsilon)$, where the tilinglike algorithm is shown to converge in six steps, i.e. $k = 6$.

We are interested in the limit of large training set sizes ($\alpha \to +\infty$). In this limit, the training error $\mathcal{E}_t(\alpha, \varepsilon)$ is close to ε:

$$\mathcal{E}_t(\alpha, \varepsilon) \simeq \varepsilon - h(\alpha, \varepsilon) \tag{50}$$

with $h(\alpha, \varepsilon)$ a function that vanishes in the limit $\alpha \to +\infty$. Note that those cost functions that do not satisfy condition (7) for all α are useless in this limit, since the error reduction at each step $\varepsilon_t^{i+1} - \varepsilon_t^i = -h(\alpha, \varepsilon_t^i)$ vanishes at some finite value of α. For larger values of α it becomes positive, and the TLA does not converge. In the preceding section we showed that the Gardner potential both with vanishing and finite margin κ has $h(\alpha, \varepsilon) > 0$ (see equations (24) and (30)) and satisfies condition (7).

As $h(\alpha, \varepsilon)$ vanishes in the limit $\alpha \to +\infty$, we can guess that the number $k(\alpha)$ diverges. In this limit we can introduce the continuum approximation, replacing i/k by the real-valued variable x. Then, the error reduction at each step is given by

$$\varepsilon_t^{i+1} - \varepsilon_t^i \simeq \frac{1}{k}\frac{\mathrm{d}\varepsilon}{\mathrm{d}x} = -h(\alpha, \varepsilon). \tag{51}$$

After integration of both sides of the equation $\mathrm{d}\varepsilon / h(\varepsilon, \alpha) = -k\,\mathrm{d}x$ at constant α, from $\varepsilon = \frac{1}{2}$ and $x = 0$ to $\varepsilon = 0$ and $x = 1$, we obtain

$$k(\alpha) \simeq \int_0^{1/2} \frac{\mathrm{d}\varepsilon}{h(\alpha, \varepsilon)} = \int_0^{1/2} \frac{\mathrm{d}\varepsilon}{\varepsilon - \mathcal{E}_t(\alpha, \varepsilon)}. \tag{52}$$

Equation (52) gives the asymptotic behaviour of the number of hidden perceptrons necessary for the TLA to converge in the limit $\alpha \to +\infty$. It depends on the cost function used to train the perceptrons through $\mathcal{E}_t(\alpha, \varepsilon)$. The storage capacity $\alpha_c(k)$ of the TLA is then obtained through the inversion of $k(\alpha)$.

Hereafter we consider the case where the hidden perceptrons are trained with the Gardner cost function, using the results of the preceding section.

We determine first the number of hidden units obtained when the perceptrons minimize the number of training errors, that is, the Gardner cost function with $\kappa = 0$. Inserting into (52), the result (24) obtained within the RS approximation, we obtain

$$k^{RS}(\alpha) \simeq \int_0^{1/2} \frac{\mathrm{d}\varepsilon}{\varepsilon - \mathcal{E}_t(\alpha, \varepsilon)} \simeq \frac{\alpha}{2\ln\alpha} \int_0^{1/2} a^2(\varepsilon)\,\mathrm{d}\varepsilon \simeq 0.475\frac{\alpha}{\ln\alpha} \tag{53}$$

where $a(\varepsilon)$ is given by (21). From this result, we deduce the storage capacity in the limit of a large number of hidden perceptrons:

$$\alpha_c^{RS}(k) \simeq 2.11k\ln k. \tag{54}$$

Surprisingly, the capacity of the TLA scales with k like the upper bound for the parity machine with the same number of hidden units, and only the prefactor is overestimated.

Using the result (45) obtained with the KT cavity method, which overestimates the perceptron's training error, we get

$$k^{KT}(\alpha) \simeq \alpha \int_0^{1/2} \frac{\mathrm{d}\varepsilon}{F(a(\varepsilon))} \simeq 1.082\alpha \tag{55}$$

where $F(a)$ is defined in (45) and $a(\varepsilon)$ is given by (43). The corresponding storage capacity is

$$\alpha_c^{KT}(k) \simeq 0.924k. \tag{56}$$

We find that $\alpha_c^{KT} < \alpha_c^{RS}$ as expected. The behaviour of the storage capacity obtained with the KT cavity method is linear in k. This suggests that including RS breaking in the replica calculation may modify the $k\ln k$ behaviour to one proportional to $k(\ln k)^\nu$ with $0 \leqslant \nu \leqslant 1$.

However, as the actual training error of the perceptrons seems closer to the RS solution than to the KT cavity result [18], we expect ν to be close to 1.

In the following we consider the parity machine obtained when the perceptrons are trained using the Gardner cost function with a finite margin κ. We get:

$$k^{RS}(\alpha, \kappa) \simeq 2\kappa^2\alpha \qquad \text{and} \qquad k^{KT}(\alpha, \kappa) \simeq \kappa\alpha\sqrt{2\ln\alpha}. \tag{57}$$

After inversion of (57), the capacities deduced within the two approximations are

$$\alpha_c^{RS}(k, \kappa) \simeq \frac{k}{2\kappa^2} \qquad \text{and} \qquad \alpha_c^{KT}(k, \kappa) \simeq \frac{k}{\kappa\sqrt{2\ln k}} \tag{58}$$

respectively. Here again, the behaviours of $k(\alpha)$ and $\alpha_c(k)$ obtained with the RS approximation and with the KT cavity method differ. In both cases, the value of κ only affects the prefactor but not the scaling with α or k. Consistently, the prefactor of α_c diverges for $\kappa \to 0$, where the expressions (57) and (58) have to be replaced by (54) and (56) respectively, as the functional dependence of the storage capacity with k is different for $\kappa = 0$.

Imposing a finite margin dramatically decreases the capacity of the TLA. More precisely, the exponents ν of the logarithmic factor differ, depending on the approximations (RS and KT cavity method), in both κ-regimes ($\nu^{RS}(\kappa = 0) = 1$, $\nu^{RS}(\kappa > 0) = 0$, $\nu^{KT}(\kappa = 0) = 0$ and $\nu^{KT}(\kappa > 0) = -\frac{1}{2}$).

It is interesting to compare the exponents determined analytically within the RS approximation, to those obtained by West and Saad [15] through a numerical iteration over the successive perceptrons' training errors. For $\kappa = 0$, they obtain ν close to 1 ($n_e = 1.070$ and 1.049, and $n_l = 1.079$ and 1.062, for $k = 1000$ and 4000 respectively (table 3 in [15])) in very good agreement with our result $\nu^{RS}(\kappa = 0) = 1$. In the case of finite κ, West and Saad find that the exponent decreases with increasing κ (figure 13 left in [15]). Our result (58) shows that the exponent does not depend on κ, only the prefactor does. The dependence found numerically is probably due to higher order corrections, that behave like $O(\exp(-2\kappa\sqrt{2\ln\alpha}+\ln\ln\alpha))$. These terms, which are less and less negligible when approaching $\kappa = 0$, hinder the determination of the power-law exponent in the asymptotic regime $\alpha \to +\infty$. Remarkably, the RS and KT exponents ν^{RS} and ν^{KT} provide correct upper and lower bounds for the exponent obtained numerically within the one-step RS breaking approximation (figure 13 right in [15]).

6. Conclusion

We determined *analytically* the typical number of hidden units needed by a simple constructive procedure, the TLA proposed in [10], to build a parity machine. The number of hidden units depends strongly on the asymptotic properties of the learning algorithm used to train them.

We showed that the cost function minimized by the hidden perceptrons has to be bounded. This rules out, in particular, the perceptron or the AdaTron learning algorithms, as with these the training error cannot decrease beyond a finite value that depends on the training set size and on the bias of the target's distribution. This is so because the hidden perceptrons have to learn highly biased output distributions. In the asymptotic regime, large thresholds are needed to minimize the training error as, loosely speaking, such solutions allow to classify correctly most patterns of the majority class. In such solutions, a non-negligible fraction of patterns have large negative stabilities. If the cost function is unbounded for $\lambda \to -\infty$, it favours solutions with small thresholds, which have large training errors. With bounded potentials, like the counting functions used in the Gardner cost function, solutions with large thresholds exist.

We deduced the properties of a perceptron with threshold, learning targets drawn with a biased distribution, trained with the Gardner cost function with and without margin. In

particular, solutions such that the training error is smaller than the bias always exist. This is a condition necessary for the TLA to converge. The asymptotic behaviour of the learning curves $\mathcal{E}_t(\alpha, \varepsilon)$ was determined through a replica calculation assuming RS, and also using the KT cavity method. The former approximation underestimates the training error, while the latter overestimates it. The main results are the expressions (24), (30), (45) and (48) relating the training error of the perceptron $\mathcal{E}_t(\alpha, \varepsilon)$ to the bias ε of the target distribution. Closer inspection of equations (24) and (30) shows that the error reduction $\mathcal{E}_t(\alpha, \varepsilon) - \varepsilon$ at large α is larger if $\kappa = 0$ than for $\kappa > 0$.

These results allow us to find analytically the number of units $k(\alpha)$ needed by the constructive procedure to converge in the large α limit. As expected, the smallest $k(\alpha)$ is obtained when the hidden perceptrons minimize their training errors, which corresponds to the Gardner cost function with $\kappa = 0$. Nevertheless, it is also worth studying the case with $\kappa > 0$, which is interesting in noisy applications. The storage capacity $\alpha_c(k)$ of the TLA is obtained through the inversion of $k(\alpha)$. Our results have been obtained under the simplifying assumption that the targets the successive perceptrons have to learn are uncorrelated. This hypothesis has been shown to be a good approximation [15] in the limit of large training sets considered here.

In the limit of large k we find $\alpha_c^{RS}(k) \simeq 2.11 k \ln k$ within the RS approximation. It is interesting to compare this algorithm-dependent storage capacity to the storage capacity of a parity machine with the same number of hidden perceptrons. The latter is independent of the learning algorithm. Geometric arguments [13] and a replica calculation where the permutation symmetry among hidden units has to be broken [14], both lead to $\alpha_c = k \ln k / \ln 2$. It is surprising that, although we disregarded the correlations between perceptrons and assumed RS, which both lead to an overestimation of the storage capacity, we find the same leading behaviour. Only the prefactor is overestimated. In fact, the permutation symmetry only arises when the perceptrons are trained simultaneously. As it is absent in the case of the incremental construction, the consequence of the RS approximation is less dramatic than in [14].

As the KT cavity method provides an upper bound to the perceptron's training error, it allows us to determine a lower bound for the TLA storage capacity. This bound scales linearly with the number of hidden units, suggesting that a calculation including full RS breaking may change the power law of the logarithmic factor. We expect that $\alpha_c \sim k(\ln k)^\nu$ with $0 \leqslant \nu \leqslant 1$.

Acknowledgment

References

[1]　Cover T M 1965 *IEEE Trans. Electron. Comput.* **14** 326–34
[2]　Gardner E 1987 *Europhys. Lett.* **4** 481–5
[3]　Gardner E 1988 *J. Phys. A: Math. Gen.* **21** 257–70
[4]　Gardner E and Derrida B 1988 *J. Phys. A: Math. Gen.* **21** 271–84
[5]　Watkin T L H, Rau A and Biehl M 1993 *Rev. Mod. Phys.* **65** 499–543
[6]　Grossman T, Meir R and Domany E 1989 *Complex Syst.* **2** 555
[7]　Gallant S I 1986 *Proc. 8th Ann. Conf. Science Soc. (Amherst, MA, August 1986)* pp 652–60
[8]　Mézard M and Nadal J-P 1989 *J. Phys. A: Math. Gen.* **22** 2191–203
[9]　Rujàn P and Marchand M 1989 *Complex Syst.* **3** 229
[10]　Biehl M and Opper M 1991 *Phys. Rev.* A **44** 6888–94
[11]　Frean M 1990 *Neural Comput.* **2** 198–209

[12] Martinez D and Estève D 1992 *Europhys. Lett.* **18** 95–100
[13] Mitchinson G J and Durbin R M 1989 *Biol. Cybern.* **60** 345
[14] Xiong Y S, Kwon C and Oh J-H 1998 *J. Phys. A: Math. Gen.* **31** 7043–9
[15] West A H L and Saad D 1998 *J. Phys. A: Math. Gen.* **31** 8977–9021
[16] Erichsen R and Theumann W K 1993 *J. Phys. A: Math. Gen.* **26** L61–8
[17] Majer P, Engel A and Zippelius A 1993 *J. Phys. A: Math. Gen.* **26** 7405–16
[18] Gerl F and Krey U 1997 *J. Physique* I **7** 303–27
[19] Abbott L F and Kepler T B 1989 *J. Phys. A: Math. Gen.* **22** L711–7
[20] Abbott L F and Kepler T B 1989 *J. Phys. A: Math. Gen.* **22** 2031–8
[21] Griniasty M and Gutfreund H 1991 *J. Phys. A: Math. Gen.* **24** 715–34
[22] Anlauf J K and Biehl M 1989 *Europhys. Lett.* **10** 687–92
[23] Gordon M B 1996 *Proc. ICNN'96* pp 381–6
[24] West A H L and Saad D 1997 *J. Phys. A: Math. Gen.* **30** 3471–96
[25] Bouten M 1994 *J. Phys. A: Math. Gen.* **27** 6021–3