Commun. Theor. Phys. **71** (2019) 243–252 Vol. 71, No. 2, February 1, 2019

Active Online Learning in the Binary Perceptron Problem*

Hai-Jun Zhou (周海军)†

CAS Key Laboratory for Theoretical Physics, Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China

School of Physical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

(Received December 6, 2018; revised manuscript received January 3, 2019)

Abstract *The binary perceptron is the simplest artificial neural network formed by N input units and one output unit, with the neural states and the synaptic weights all restricted to ± 1 values. The task in the teacher-student scenario is to infer the hidden weight vector by training on a set of labeled patterns. Previous efforts on the passive learning mode have shown that learning from independent random patterns is quite inefficient. Here we consider the active online learning mode in which the student designs every new Ising training pattern. We demonstrate that it is mathematically possible to achieve perfect (error-free) inference using only N designed training patterns, but this is computationally unfeasible for large systems. We then investigate two Bayesian statistical designing protocols, which require $2.3N$ and $1.9N$ training patterns, respectively, to achieve error-free inference. If the training patterns are instead designed through deductive reasoning, perfect inference is achieved using $N+\log_2 N$ samples. The performance gap between Bayesian and deductive designing strategies may be shortened in future work by taking into account the possibility of ergodicity breaking in the version space of the binary perceptron.*

DOI: 10.1088/0253-6102/71/2/243

Key words: neural network, perceptron, online learning, belief propagation, statistical inference

1 Introduction

The perceptron invented by Frank Rosenblatt in 1957 is probably the simplest artificial neural network.[1] It has N inputs and one output, with each input neuron i affecting the output neuron through a synapse of weight T_i.[2−3] Given an N-dimensional input vector $\boldsymbol{\xi} \equiv (\xi_1, \xi_2, \ldots, \xi_N)$, the binary output state σ is determined according to a (highly nonlinear) sign function

$$\sigma(\boldsymbol{\xi}) = \mathrm{sign}(\boldsymbol{\xi}; \boldsymbol{T}) \equiv \mathrm{sign}\Big(\sum_{i=1}^{N} T_i \xi_i\Big)\,, \tag{1}$$

where $\boldsymbol{T} \equiv (T_1, T_2, \ldots, T_N)$ denotes the synaptic weight vector. The output $\sigma = 1$ if the overlap between $\boldsymbol{\xi}$ and $\boldsymbol{T}$, defined as $\sum_i T_i s_i$, is positive and $\sigma = -1$ if this overlap is negative. We consider the binary (or Ising) perceptron problem, so all the synaptic weights and neural states are restricted to be Ising-valued (i.e., $T_i, \xi_i \in \pm 1$ for every input neuron i of the system). These constraints make the binary perceptron much more challenging to study than the continuous counterpart.[2−3]

The perceptron can serve as a linear classifier. In this scenario, given P patterns $\boldsymbol{\xi}^\mu \equiv (\xi_1^\mu, \xi_2^\mu, \ldots, \xi_N^\mu)$ with indices $\mu = 1, 2, \ldots, P$ and their binary labels σ^μ, the task is to find a vector $\boldsymbol{T}$ such that these P patterns are correctly classified, that is, $\sigma^\mu = \mathrm{sign}(\boldsymbol{\xi}^\mu; \boldsymbol{T})$ holds for each and every one of these P patterns.[2−3] The whole set of all such compatible weight vectors form the version space (or solution space) of the perceptron. If the input binary patterns $\boldsymbol{\xi}^\mu$ are random and independent, it was predicted using the replica method of statistical physics that there exist binary solutions $\boldsymbol{T}$ to this classification problem as long as $P < 0.83N$.[4] Several message-passing algorithms[5−6] inspired by the cavity method of statistical physics have been implemented to solve single instances of the perceptral classification problem. More recently it was revealed that, as the density $\alpha \equiv P/N$ of random input binary patterns increases, the typical (equilibrium) solutions of this classification problem become widely separated from each other and are extremely hard to reach,[7−8] while there are also sub-dominant dense regions in the version space, which could be reached through entropy-weighted sampling strategies.[9−10]

The perceptron can also be studied from the teacher–student perspective, with $\boldsymbol{T}$ understood as the teacher's weight vector which is hidden to the student. For each query $\boldsymbol{\xi}$ to the teacher, the correct label $\sigma(\boldsymbol{\xi})$ as computed through Eq. (1) is revealed to the student, and the task for the student is to infer the hidden vector $\boldsymbol{T}$ by learning from the $(\boldsymbol{\xi}, \sigma)$ associations.[2−3] This inference task may be carried out in an offline manner, meaning that the training samples are repeatedly examined by the student during the learning process. It may also be carried out in an online manner, meaning that each training

*Supported by the National Natural Science Foundation of China under Grant Nos. 11421063 and 11747601 and the Chinese Academy of Sciences under Grant No. QYZDJ-SSW-SYS018

†Corresponding author, E-mail: zhouhj@itp.ac.cn

sample is used only once by the student to update his/her belief on $\boldsymbol{T}$. We study online learning in the present work.

The learning performance of the teacher–student perceptron system has been investigated by many authors. In the passive learning mode for which the training patterns are independent and random, it was predicted that perfect (error-free) inference of a binary vector $\boldsymbol{T}$ is theoretically possible with $P \approx 1.25N$ binary samples.[11−12] But this theoretical limit has never been achieved by actual heuristic algorithms. Theoretical and numerical studies on various algorithms have found that the generalization error ε of the passive learning process decreases with the pattern density α algebraically, e.g., $\varepsilon \propto \alpha^{-1}$. This means that in the thermodynamic limit of $N \to \infty$, perfect inference is unlikely to achieve at any finite value of pattern density α.[13−19]

In this work we address the issue of active learning, which aims at accelerating online inference by carefully designing the training patterns. After the student has encountered P samples and has already gained some knowledge about the truth vector $\boldsymbol{T}$, how should she/he design the $(P+1)$-th query so that the answer from this new query will be most informative for inference? This interesting question was explored by many authors in the early 1990s (see, e.g., Refs. [20–27]), but the focus was on minimization of the generalization error rather than on error-free inference. In Sec. 2 of the present paper we prove that error-free learning of $\boldsymbol{T}$ can be achieved using at most $N+\log_2 N$ designed training patterns through deductive reasoning, which is only slightly beyond the theoretical lower bound, N. If the optimal Bayesian inference strategy is used instead of deductive logic, we find that error-free inference using exactly N designed samples is indeed possible, but it is computationally feasible only for small systems (Sec. 3). We then implement two heuristic designing algorithms in Secs. 4 and 5 for large systems based on this optimal Bayesian principle. Our simulation results demonstrate that these two heuristic algorithms need $2.3N$ and $1.9N$ training samples, respectively, to achieve error-free inference.

Although the deductive-logic algorithm certainly outperforms the data-driven Bayesian algorithms, the observation that the Bayesian statistical approach achieves perfect inference of N bits with less than $2N$ one-bit measurements is still quite encouraging. We expect that the performance of the Bayesian active inference algorithms will be further improved after taking into account the possibility of ergodicity breaking in the version space of the perceptron. If the version space divides into a large number of well-separated clusters, the assumption of Gaussian distribution of the mean field theory will no longer be valid (Sec. 6), and more advanced mean field theories such as the fist-step replica-symmetry-breaking cavity method will be needed to better describe the complicated statistical correlations of the version space.

The concept of active (or adaptive) learning has been widely discussed in the fields of education science[28] and optimal experimental design.[29−30] Science itself may also be considered as an active learning process,[29] for which data-inspired intuitive insights, controlled experiments, and deductive reasoning are all indispensable. Artificial deep neural networks are becoming powerful tools for extracting the most important features from huge amount of data,[31−32] facilitating hypothesis formation and experimental design. Recently there has been great enthusiasm in this direction, and efficient active learning algorithms for deep neural networks are start to be explored.[33−36] A lot of work remains to be done on this important issue. From the theoretical side, the binary perceptron may serve as a simple model system to push for the limit of active Bayesian learning. Another basic inference problem which is closely related to the perceptron model is the so-called one-bit compressed sensing problem.[37−38] At the moment only the passive mode of one-bit compressed sensing has been considered in the literature. Beyond the single-layered perceptron and one-bit compressed sensing, the next and more challenging model is the multi-layered binary perceptron system.

2 Inference by Deductive Reasoning

We first show that if the student employs deductive reasoning, online perceptral learning can be made very efficient. At the start of the learning process, the student can simply choose an arbitrary pattern $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_N) \in \{-1, +1\}^N$ as the query. For convenience of discussion, let us define a particular overlap function $q(n)$ on the integer domain $n \in \{0, 1, \ldots, N\}$ as

$$q(n) = \begin{cases} \sum\limits_{i=1}^{N} \xi_i T_i \,, & (n = 0)\,, \\ -\sum\limits_{i=1}^{n} \xi_i T_i + \sum\limits_{j=n+1}^{N} \xi_j T_j \,, & (1 \le n < N)\,, \\ -\sum\limits_{i=1}^{N} \xi_i T_i \,, & (n = N)\,. \end{cases} \tag{2}$$

This function is simply the overlap (or the scalar product) of the teacher's weight vector $\boldsymbol{T}$ and the modified pattern $\boldsymbol{\xi}(n) \equiv (-\xi_1, \ldots, -\xi_n, \xi_{n+1}, \ldots, \xi_N)$ after flipping the first n entries of $\boldsymbol{\xi}$. An example of function $q(n)$ is illustrated in Fig. 1. Because N is odd, $q(n)$ takes only odd values. And because $q(N) = -q(0)$ and $|q(n) - q(n+1)| = 2$ for any $n < N$ (quasi-continuity), there must exist at least one $n^* \in \{0, 1, \ldots, N-1\}$ for which $|q(n^*)| = |q(n^*+1)| = 1$ and $q(n^*+1) = -q(n^*)$.

The value of such an integer n^* can be determined through at most $\log_2 N$ queries. Starting from $n_l = 0$ and $n_r = N$, the query sequence goes as follows: (i) feed pattern $\boldsymbol{\xi}(n_l)$ to the perceptron to get the sign of $q(n_l)$; (ii)

set $n_m \equiv \lceil (n_l + n_r)/2 \rceil$ and feed pattern $\boldsymbol{\xi}(n_m)$ to the perceptron to get the sign of $q(n_m)$; (iii) if $q(n_m)$ and $q(n_l)$ have the same sign, set $n_l = n_m$, otherwise set $n_r = n_m$; (iv) if $n_r = n_l + 1$, set $n^* = n_l$ and quit, otherwise repeat steps (ii)–(iv).

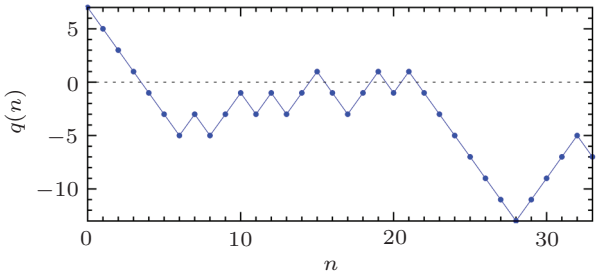

Fig. 1 An example of the overlap function $q(n)$ for a small perceptron of size $N = 33$. The random initial binary pattern $\boldsymbol{\xi}$ happens to have overlap $q(0) = 7$ with the teacher's binary weight vector $\boldsymbol{T}$, and $q(n)$ is the new overlap with $\boldsymbol{T}$ after the first n entries of $\boldsymbol{\xi}$ are all flipped. In this example $q(n)$ changes sign seven times.

The value of $q(n^*)$, i.e. the overlap between $\boldsymbol{\xi}(n^*)$ and $\boldsymbol{T}$, must be $+1$ or -1. Because of this fact, if one flips the i-th entry of $\boldsymbol{\xi}(n^*)$ and feeds the resulting slightly modified pattern to the perceptron, the value of T_i can be deduced from the output. That is, if the output is the same as the sign of $q(n^*)$, then T_i must be equal to $-\xi_i^* q(n^*)$, otherwise T_i must be equal to $\xi_i^* q(n^*)$.

By the above-mentioned deductive reasoning method, the binary weight vector $\boldsymbol{T}$ can be exactly determined after at most $N + \log_2 N$ queries. But it is clear from the above discussions that the success of this logical approach requires a deep understanding of the perceptron system. We continue to explore other active learning strategies in the next three sections.

3 Version Space Minimization

After P training samples $(\boldsymbol{\xi}^\mu, \sigma^\mu)$ have been experienced, for a weight vector $\boldsymbol{J} = (J_1, J_2, \ldots, J_N) \in \{-1, +1\}^N$ to be compatible with all these P samples, it must satisfy

$$\mathrm{sign}\Big(\sum_{i=1}^{N} J_i \xi_i^\mu\Big) = \sigma^\mu \,, \tag{3}$$

for every one of these samples. We denote by Σ_P the set of all the Ising weight vectors $\boldsymbol{J}$ satisfying these P constraints, and refer to this set as the version space at stage P. Notice the teacher's vector $\boldsymbol{T}$ is always a member of Σ_P, so the volume $\big|\Sigma_P\big|$ of the version space must be positive-definite. To accelerate online learning, a simple idea would be to reduce the volume of the version space as much as possible with each new training pattern. When finally the version space shrinks to a single point, this surviving element must be $\boldsymbol{T}$. Given the version space Σ_P at stage P, then how should the student construct the next, $(P+1)$-th, Ising training pattern $\boldsymbol{\xi}^{P+1}$?

Consider two vectors $\boldsymbol{J}$, $\boldsymbol{J}'$ of Σ_P. Each of them has equal probability to be the truth vector $\boldsymbol{T}$ (the uniform Bayesian prior distribution is assumed). If $\boldsymbol{J}$ happens to be the truth, then the other vector $\boldsymbol{J}'$ can be refuted by a test pattern $\boldsymbol{\xi}$ if $\mathrm{sign}(\boldsymbol{\xi}; \boldsymbol{J}') = -\mathrm{sign}(\boldsymbol{\xi}; \boldsymbol{J})$. The probability of a randomly chosen vector $\boldsymbol{J}' \in \Sigma_P$ being refuted by the test pattern $\boldsymbol{\xi}$ is then

$$\frac{1}{2}\Big[1 - \frac{\mathrm{sign}(\boldsymbol{\xi}; \boldsymbol{J}) \sum_{\boldsymbol{J}' \in \Sigma_P} \mathrm{sign}(\boldsymbol{\xi}; \boldsymbol{J}')}{|\Sigma_P|}\Big] . \tag{4}$$

On the other hand, every $\boldsymbol{J} \in \Sigma_P$ is equally likely to be the truth $\boldsymbol{T}$, so we need to maximize the mean value of the above expression over all the different choices of $\boldsymbol{J}$,

$$\frac{1}{2}\Big[1 - \frac{\sum_{\boldsymbol{J} \in \Sigma_P} \mathrm{sign}(\boldsymbol{\xi}; \boldsymbol{J})}{|\Sigma_P|} \frac{\sum_{\boldsymbol{J}' \in \Sigma_P} \mathrm{sign}(\boldsymbol{\xi}; \boldsymbol{J}')}{|\Sigma_P|}\Big] , \tag{5}$$

which leads to the following constraint

$$\sum_{\boldsymbol{J} \in \Sigma_P} \mathrm{sign}(\boldsymbol{\xi}^{P+1}; \boldsymbol{J}) = 0 \,. \tag{6}$$

In other words, $\boldsymbol{\xi}^{P+1}$ should be designed to divide the old version space Σ_P into two parts of (almost) equal size. This designed pattern $\boldsymbol{\xi}^{P+1}$ must not be completely random, since the corresponding sum (6) for a completely random pattern will be of order $\sqrt{|\Sigma_P|}$. After $\boldsymbol{\xi}^{P+1}$ has been examined, one half of the members of Σ_P will be discarded and the surviving vectors form the new version space Σ_{P+1}. Let us remark that this idea of version space bisection began to be discussed in the mathematics community in the early 1970s,[39] and it is underlying the widely appreciated optimal (minimum-error) Bayesian classification algorithm for the passive perceptron problem.[14,40–41]

We test the performance of this conceptually simple designing principle on small perceptrons of size $N \leq 25$, for which the whole version space can be stored in the memory of a desktop computer. In Fig. 2, we show how the entropy density s and the generalization error ε change with the density α of training patterns. The entropy density measures the volume of the version space, $s \equiv (1/N) \log_2 |\Sigma_P|$. The generation error is computed as the probability that a randomly sampled binary test pattern will be mis-classified by a randomly chosen member $\boldsymbol{J}$ of the version space.[2–3] If the training patterns are randomly and independently drawn from the configuration space $\{-1, +1\}^N$ (i.e., the passive learning mode), the mean entropy density s and mean generalization error ε both decrease gradually with α and are positive when α exceeds unity. On the other hand if the training patterns are required to satisfy Eq. (6) we find that the mean entropy density s decreases linearly from $s = 1$ to $s = 0$ as α increases from 0 to 1, and at $\alpha = 1$ the generalization error ε becomes exactly zero. In other words, perfect inference

of the N-dimensional Ising vector $\boldsymbol{T}$ is achieved by the active strategy (6) with only N one-bit queries (which only return ± 1 values). Compared with the deductive reasoning approach, $\log_2 N$ training patterns are saved following the designing principle (6). No algorithms can do better than this Bayesian strategy, because at least N measurements are needed to exactly fix an N-dimensional vector.

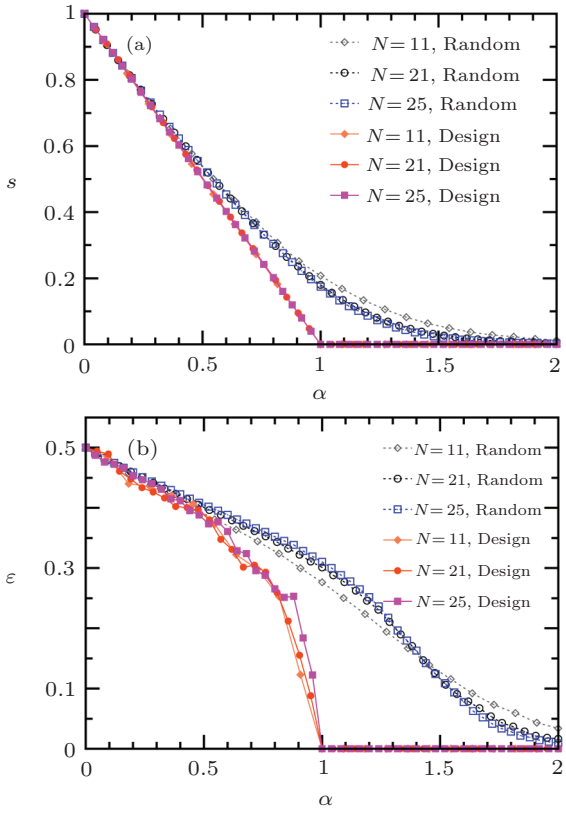

Fig. 2 The active learning strategy (6) outperforms passive learning on small Ising perceptrons: (a) entropy density s (in units of bit), (b) generalization error ε. Training patterns are added one after another, $\alpha = P/N$ is the instantaneous density of patterns (N is the number of input neurons and P is the number of training patterns). Each data point is obtained by averaging over 1,000 independent runs of the passive (Random) or the active (Design) learning algorithm.

These simulation results indicate that, in principle, perfect learning using only N training patterns is possible. But directly employing Eq. (6) to construct new training patterns is practically feasible only for small systems. When the dimension N becomes large the version space Σ_P (for P small) will be too huge to enumerate. We must convert Eq. (6) into a form suitable for implementation in large systems. This issue is addressed in the remaining part of this section.

With respect to all the accumulated P training patterns at the end of the P-th learning stage, the volume of the version space Σ_P (the partition function) is expressed as

$$|\Sigma_P| = \sum_{\boldsymbol{J}} \prod_{\mu=1}^{P} \Theta\Big(\sigma^\mu \sum_{j=1}^{N} J_j \xi_j^\mu\Big)\,, \tag{7}$$

where σ^μ is the true label of pattern $\boldsymbol{\xi}^\mu$, and $\Theta(x)$ is the Heaviside step function such that $\Theta(x) = 1$ for $x > 0$ and $\Theta(x) = 0$ for $x \leq 0$. The probability $\mathcal{P}_P(\boldsymbol{J})$ of weight vectors $\boldsymbol{J}$ in Σ_P is

$$\mathcal{P}_P(\boldsymbol{J}) = \frac{1}{|\Sigma_P|} \prod_{\mu=1}^{P} \Theta\Big(\sigma^\mu \sum_{j=1}^{N} J_j \xi_j^\mu\Big)\,. \tag{8}$$

Notice that this probability distribution depends on the details of the P training patterns. If $\boldsymbol{J} \in \Sigma_P$ we have $\mathcal{P}_P(\boldsymbol{J}) = 1/|\Sigma_P|$, otherwise $\mathcal{P}_P(\boldsymbol{J}) = 0$. Besides this joint distribution of all the N entries J_i of $\boldsymbol{J}$, we are also interested in single-weight marginals. The mean value of weight J_i among all the vectors of Σ_P is

$$\langle J_i \rangle_P \equiv \frac{1}{|\Sigma_P|} \sum_{\boldsymbol{J} \in \Sigma_P} J_i = \sum_{\boldsymbol{J}} \mathcal{P}_P(\boldsymbol{J}) J_i\,. \tag{9}$$

Similarly, the mean value of $J_i J_j$ (pair correlation) is

$$\langle J_i J_j \rangle_P \equiv \frac{1}{|\Sigma_P|} \sum_{\boldsymbol{J} \in \Sigma_P} J_i J_j = \sum_{\boldsymbol{J}} \mathcal{P}_P(\boldsymbol{J}) J_i J_j\,. \tag{10}$$

When $i = j$ we have $\langle J_i J_i \rangle_P = 1$ due to the Ising nature of the weights. The training patterns bring correlations among the different weight variables. A consequence of these complicated correlations is that $\langle J_i J_j \rangle_P \neq \langle J_i \rangle_P \langle J_j \rangle_P$ for $i \neq j$.

Now consider adding a new training pattern $\boldsymbol{\xi}$ to the perceptron. The distribution $\mathcal{P}_P(q|\boldsymbol{\xi})$ of the overlap q between $\boldsymbol{\xi}$ and the weight vectors $\boldsymbol{J}$ of Σ_P is defined as

$$\mathcal{P}_P(q|\boldsymbol{\xi}) = \sum_{\boldsymbol{J}} \mathcal{P}_P(\boldsymbol{J}) \delta\Big(q - \sum_{i=1}^{N} \xi_i J_i\Big)\,, \tag{11}$$

where $\delta(x)$ is the Dirac symbol such that $\delta(x) = 1$ for $x = 0$ and $\delta(x) = 0$ for $x \neq 0$. From this definition we see that the mean overlap $q(\boldsymbol{\xi}) \equiv \int \mathrm{d}q \mathcal{P}_P(q|\boldsymbol{\xi}) q$ is

$$q(\boldsymbol{\xi}) = \sum_{i=1}^{N} \xi_i \langle J_i \rangle_P\,. \tag{12}$$

The variance of the overlap q is defined as $\Delta(\boldsymbol{\xi}) \equiv \int \mathrm{d}q \mathcal{P}_P(q|\boldsymbol{\xi}) q^2 - [q(\boldsymbol{\xi})]^2$. By a simple derivation we find that

$$\Delta(\boldsymbol{\xi}) = \sum_i (1 - \langle J_i \rangle_P^2) + \sum_{i<j} 2\xi_i \xi_j (\langle J_i J_j \rangle_P - \langle J_i \rangle_P \langle J_j \rangle_P)\,. \tag{13}$$

The overlap q is the sum of N random terms $\xi_i J_i$ (randomness coming from J_i). As the lowest-order approximation we assume that the central limit theorem is valid for q when N is large, even through the weights J_i are not independent. In other words, we approximate the probability

distribution (11) by a Gaussian distribution:

$$\mathcal{P}_P(q|\boldsymbol{\xi}) \approx \frac{1}{\sqrt{2\pi\Delta(\boldsymbol{\xi})}} \exp\Big(-\frac{\big(q-q(\boldsymbol{\xi})\big)^2}{2\Delta(\boldsymbol{\xi})}\Big)\,. \tag{14}$$

(The possible breaking down of this Gaussian assumption will be discussed in Sec. 6)

According to the designing principle (6), the $(P+1)$-th training pattern should refute half of the weight vectors in Σ_P. This means that the overlap between $\boldsymbol{\xi}^{P+1}$ and the weight vectors of Σ_P should be positive for half of the elements $\boldsymbol{J} \in \Sigma_P$ and be negative for the remaining half. According to the Gaussian approximation (14), we see that the new pattern $\boldsymbol{\xi}^{P+1}$ should have zero mean overlap value with the weight vectors of Σ_P, that is

$$\sum_{i=1}^{N} \xi_i^{P+1} \langle J_i\rangle_P = 0\,. \tag{15}$$

In comparison with Eq. (6), the advantage of Eq. (15) is that the student does not need to memorize all the candidate truth vectors $\boldsymbol{J}$ but only need to evaluate the N mean synaptic weights $\langle J_i\rangle_P$. Equation (15) may be regarded as a linearized version of Eq. (6) with the highly nonlinear sign function replaced by a linear function. In the next section we discuss how to efficiently update the mean weights during the online learning process. We notice that the constraint (15) is very similar in form to the designing constraint discussed in some of the early papers.[20,25] A significant difference is that our proposed constraint (15) involves the mean synaptic weights $\langle J_i\rangle_P$, instead of the synaptic weights J_i of a single weight vector stored in memory.

We employ simulated annealing[42] to sample a maximally random pattern $\boldsymbol{\xi}^{P+1}$ under constraint (15). An energy penalty is defined for each Ising pattern $\boldsymbol{\xi}$ as

$$E(\boldsymbol{\xi}) = \Big|\sum_{i=1}^{N} \langle J_i\rangle_P \xi_i\Big|\,, \tag{16}$$

and the corresponding probability distribution of $\boldsymbol{\xi}$ is

$$\mathcal{P}(\boldsymbol{\xi}) \propto \exp(-\beta E(\boldsymbol{\xi})) = \exp\Big(-\beta|\sum_{i=1}^{N} \langle J_i\rangle_P \xi_i|\Big)\,, \tag{17}$$

where the parameter β is the inverse temperature. The pattern $\boldsymbol{\xi}$ evolves by single spin flips at slowly increasing β values. At each elementary update step, (i) a randomly chosen entry ξ_i of $\boldsymbol{\xi}$ is flipped to the opposite value; (ii) if the energy difference $\Delta E \equiv E(\boldsymbol{\xi}') - E(\boldsymbol{\xi})$ between the modified pattern $\boldsymbol{\xi}'$ and the old pattern $\boldsymbol{\xi}$ is non-positive, then this flip $\xi_i \to -\xi_i$ is surely accepted, otherwise it is accepted only with probability $\mathrm{e}^{-\beta\Delta E}$. After N such elementary flip trials the value of β is then elevated by a constant factor r_β ($\beta \leftarrow r_\beta\beta$). Finally, the spin configuration after a total number tN of spin flip trials (β has been recursively elevated t times) is picked as the new training pattern $\boldsymbol{\xi}^{P+1}$. In this work we set $r_\beta = 1.1$ and $t = 100$, and set the initial value of β to be 0.01. We have checked that the numerical results of the next two sections are insensitive to the particular values of these parameters.

4 Experience Accumulation

To exploit the designing principle (15) we must first compute $\langle J_i\rangle_P$ for all the weight indices i. At $P = 0$ we know that $\langle J_i\rangle_0 = 0$ for all the synaptic weights J_i. But the task for $P \geq 1$ is quite non-trivial and can not be made exact. In the online learning paradigm we compute $\langle J_i\rangle_P$ approximately by iteration.

After the training pattern $\boldsymbol{\xi}^{P+1}$ has been added to the system, the new distribution $\mathcal{P}_{P+1}(\boldsymbol{J})$ of the version space is related to the old $\mathcal{P}_P(\boldsymbol{J})$ through

$$\mathcal{P}_{P+1}(\boldsymbol{J}) = \frac{\Theta\Big(\sigma^{P+1}\sum_j \xi_j^{P+1} J_j\Big)\mathcal{P}_P(\boldsymbol{J})}{\sum_{\boldsymbol{J}'}\Theta\Big(\sigma^{P+1}\sum_j \xi_j^{P+1} J_j'\Big)\mathcal{P}_P(\boldsymbol{J}')}\,, \tag{18}$$

where $\sigma^{P+1} \equiv \mathrm{sign}\big(\sum_i T_i \xi_i^{P+1}\big)$ is the label of $\boldsymbol{\xi}^{P+1}$. The mean value of J_i under $\mathcal{P}_{P+1}(\boldsymbol{J})$ is

$$\langle J_i\rangle_{P+1} = \frac{p_i^+ A_i^+ - p_i^- A_i^-}{p_i^+ A_i^+ + p_i^- A_i^-}\,. \tag{19}$$

Here, $p_i^+ \equiv (1 + \langle J_i\rangle_P)/2$ is the probability of $J_i = +1$ in the version space Σ_P, and $p_i^- \equiv 1 - p_i^+$ is the complementary probability; A_i^+ and A_i^- are, respectively, the conditional probabilities of $\boldsymbol{\xi}^{P+1}$ being correctly classified by a weight vector $\boldsymbol{J} \in \Sigma_P$ given that $J_i = +1$ and $J_i = -1$. The defining expressions for A_i^+ and A_i^- are

$$A_i^+ \equiv \frac{1}{|\Sigma_P^{i+}|}\sum_{\boldsymbol{J}\in\Sigma_P^{i+}} \Theta\Big(\sigma^{P+1}\xi_i^{P+1} + \sigma^{P+1}\sum_{j\neq i} J_j\xi_j^{P+1}\Big)\,, \tag{20a}$$

$$A_i^- \equiv \frac{1}{|\Sigma_P^{i-}|}\sum_{\boldsymbol{J}\in\Sigma_P^{i-}} \Theta\Big(-\sigma^{P+1}\xi_i^{P+1} + \sigma^{P+1}\sum_{j\neq i} J_j\xi_j^{P+1}\Big)\,, \tag{20b}$$

where Σ_P^{i+} (respectively, Σ_P^{i-}) denotes the version subspace of Σ_P which contains all the vectors $\boldsymbol{J} \in \Sigma_P$ with $J_i = +1$ (respectively, $J_i = -1$). Notice that $\Sigma_P^{i+} \cap \Sigma_P^{i-} = \emptyset$ and $\Sigma_P^{i+} \cup \Sigma_P^{i-} = \Sigma_P$. Similar to the derivation of Eq. (14) we can approximate A_i^+ and A_i^- by two slightly different Gaussian integrals (details in Appendix A). Then we get the following convenient iterative equation for the mean weights

$$\langle J_i\rangle_{P+1} \approx \langle J_i\rangle_P + \sigma^{P+1}\xi_i^{P+1}(1 - \langle J_i\rangle_P^2)R_P\,. \tag{21}$$

Here R_P is a magnitude factor determined by

$$R_P \equiv \frac{2}{\sqrt{2\pi\Delta(\boldsymbol{\xi}^{P+1})}} f\Big(\frac{\sigma^{P+1}\sum_{k=1}^{N}\xi_k^{P+1}\langle J_k\rangle_P}{\sqrt{2\Delta(\boldsymbol{\xi}^{P+1})}}\Big)\,, \tag{22}$$

with the function $f(x)$ being

$$f(x) \equiv \frac{\exp(-x^2)}{1 + (2/\sqrt{\pi})\int_0^x \exp(-t^2)\mathrm{d}t}\,. \tag{23}$$

The value of $f(x)$ equals to unity at $x = 0$, and it then rapidly decays to zero as x increases. When x is negative $f(x)$ rapidly approaches the asymptotic form $-2x$.

The iterative expression (21) agrees with the belief-propagation equation reported in Ref. [6] (see also Refs. [5,43]). Notice that if ξ_i^{P+1} has the same (respectively, opposite) sign of σ^{P+1}, the value of $\langle J_i\rangle_{P+1}$ increases (respectively, decreases) from $\langle J_i\rangle_P$ by an amount $(1-\langle J_i\rangle_P^2)R_P$. Equation (21) therefore implements a specific Hebbian rule of experience accumulation. Notice also that $\langle J_i\rangle_P = \pm 1$ are two fixed points of Eq. (21), so if $\langle J_i\rangle_P$ is wrongly estimated to be (say) -1 while the truth value is $T_i = +1$, there is no chance to correct this mistake by further learning. To ensure that the iteration process is able to escape from a wrong fixed point, we therefore slightly modify Eq. (21) as follows:

$$\langle J_i\rangle_{P+1} = \langle J_i\rangle_P + \sigma^{P+1}\xi_i^{P+1} W\big(\langle J_i\rangle_P\big) R_P\,, \qquad (24)$$

where the weighting function $W\big(\langle J_i\rangle_P\big)$ by default is equal to $1-\langle J_i\rangle_P^2$, but $W\big(\langle J_i\rangle_P\big) = W_0$ if the sign of $\sigma^{P+1}\xi_i^{P+1}$ is opposite to that of $\langle J_i\rangle_P$ and at the same time $1-\langle J_i\rangle_P^2 < W_0$. The precise value of the cutoff parameter has a weak effect on the learning performance. After some preliminary experiments we finally set $W_0 = 8\times 10^{-3}$.

To determine the numerical value of the magnitude factor R_P, we need to compute the numerical value of the overlap variance $\Delta(\boldsymbol{\xi}^{P+1})$. The first summation of Eq. (13) is of order N and is easy to compute. On the other hand, it is quite complicated to compute the second summation of Eq. (13). In this work we make the simplest approximation of independence among different weight variables, that is, $\langle J_i J_j\rangle_P \approx \langle J_i\rangle_P\langle J_j\rangle_P$. Under this additional approximation then

$$\Delta(\boldsymbol{\xi}^{P+1}) \approx \sum_{i=1}^{N}(1-\langle J_i\rangle_P^2)\,, \qquad (25)$$

and it is completely independent of $\boldsymbol{\xi}^{P+1}$. This independence approximate expression is commonly adopted in the literature (see, e.g., Refs. [6,44]), it amounts to approximate the probability distribution $\mathcal{P}_P(\boldsymbol{J})$ by a factorized form. To improve the numerical accuracy of computing the mean weight values, the next step is to include at least partially the correlations between the weight elements (see, e.g., discussions in Refs. [6,44] concerning the continuous Perceptron). We leave this demanding issue for future work. (More discussion is made in Sec. 6 on weight correlations.)

We now test the performance of the simple designing principle (15). The following straightforward inference rule is adopted in the computer simulations: At the end of the P-th learning stage, the inferred truth vector $\hat{\boldsymbol{T}}^P \equiv (\hat{T}_1^P, \hat{T}_2^P, \ldots, \hat{T}_N^P)$ is simply the sign vector of the mean weights $\langle J_i\rangle_P$, with

$$\hat{T}_i^P = \mathrm{sign}\big(\langle J_i\rangle_P\big)\,. \qquad (26)$$

The relative inference error is defined as the relative Hamming distance between the inferred weight vector $\hat{\boldsymbol{T}}^P$ and the teacher's weight vector $\boldsymbol{T}$, that is

$$d(\hat{\boldsymbol{T}}^P, \boldsymbol{T}) \equiv \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{T}_i^P - T_i|}{2}\,. \qquad (27)$$

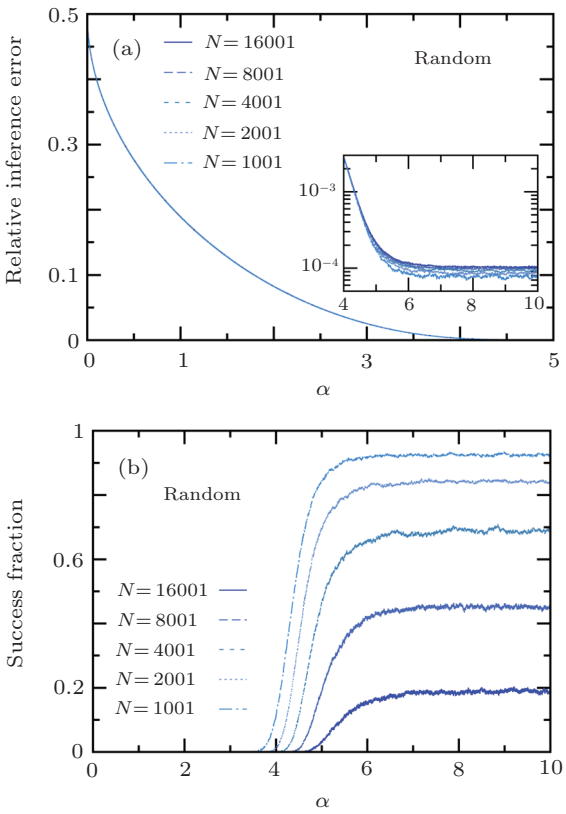

Fig. 3 The performance of passive online learning. The P training patterns are fed to the student sequentially and they are independent random N-dimensional Ising vectors. The pattern density is $\alpha = P/N$. The total number of simulated independent online learning trajectories is $\mathcal{N} = 10^4$. (a) The mean inference error, i.e., the mean fraction of incorrectly inferred teacher weights. The inset shows the tail part of the numerical data in semi-logarithmic scale. (b) The success fraction, i.e., the fraction of simulation trajectories in which the inferred weight vector is identical to the teacher's weight vector.

Before testing the active learning mode, we first consider the random passive learning mode, in which every newly introduced training pattern is drawn independently and uniformly at random from the set of 2^N Ising patterns. The mean value of the relative inference error of this passive mode, averaged over $\mathcal{N} = 10^4$ simulated independent online learning trajectories, is shown in Fig. 3(a) as a function of pattern density α. We find that when $\alpha < 5$, the curves of mean inference error for different system sizes N are well superimposed onto each other, while for $\alpha \geq 5$ the mean inference error saturates to a small positive level whose height slightly increases with system size N (see inset of Fig. 3(a)).

As another measure of performance we consider the success fraction, which is defined as the probability that the inferred weight vector $\hat{\boldsymbol{T}}^P$ is identical to the truth vector $\boldsymbol{T}$ in repeated independent run of the whole online learning process. For example, if $\hat{\boldsymbol{T}}^P = \boldsymbol{T}$ in 4000 out of 10,000 independent runs then the success fraction is 0.4 at this particular pattern density $\alpha = P/N$. Notice that the success fraction is not necessarily a monotonic function of α because it is possible that $\hat{\boldsymbol{T}}^P = \boldsymbol{T}$ but $\hat{\boldsymbol{T}}^{P+1} \neq \boldsymbol{T}$. The success fraction of passive online learning is shown in Fig. 3(b) as a function of pattern density α. We find this fraction is first identical to zero for small α values, it then increases quickly when α exceeds 4 and finally fluctuates around a plateau level. Since the height of the plateau level decreases considerably with system size N, error-free inference will be impossible in the thermodynamic limit of $N \to \infty$.

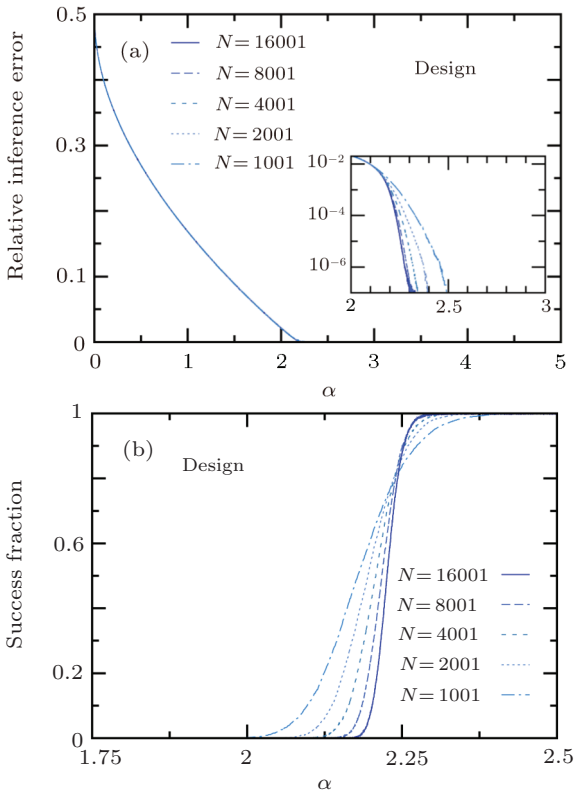

Fig. 4 Same as Fig. 3, but for active online learning under the designing principle (15). Error-free inference is achieved at pattern density $\alpha \approx 2.23$.

When constraint (15) is imposed in designing each new training pattern, we find that the learning performance is greatly enhanced. As shown in Fig. 4(a), the mean inference error reaches below 10^{-3} as the pattern density α is increased up to 2.2 and it then quickly drops to zero as α further increases slightly. The dramatic effect of active learning is most clearly demonstrated by the difference between Fig. 4(b) and Fig. 3(b). We see from Fig. 4(b) that the curve of success fraction becomes more and more sharper as the system size N increases, and all these different curves intersect at approximately the same value of α. Similar system size-dependent behaviors are commonly observed in finite-size scaling studies of continuous phase transitions.[45] We conjecture that a well-defined dynamical phase-transition to perfect inference will occur at the value of $\alpha \approx 2.23$ in the thermodynamic limit of $N \to \infty$. More thorough theoretical investigation on the large N limit of this learning dynamics will be carried in a follow-up paper.

5 Additional Orthogonality Considerations

When a new training pattern $\boldsymbol{\xi}^{P+1}$, constrained by Eq. (15) but otherwise being maximally random, is sampled by simulated annealing (Eq. (17)), it is conditionally independent of all the earlier training patterns given the values of $\langle J_j \rangle_P$. But since the mean weights $\langle J_j \rangle_P$ are determined through the accumulative mechanism (24), Eq. (15) indeed brings complicated correlations between $\boldsymbol{\xi}^{P+1}$ and all its predecessors. If $\boldsymbol{\xi}^{P+1}$ happens to be relatively similar to some of the old training patterns, its power in promoting active inference will be compromised.[23,46] According to the geometric picture underlying the exact designing principle (6), it should be beneficial to explicitly require (at least approximate) orthogonality between $\boldsymbol{\xi}^{P+1}$ and the training patterns introduced during the last M steps.

To implement these additional orthogonality constraints, we modify the energy function of the simulated annealing process as follows:

$$E(\boldsymbol{\xi}) = \Big|\sum_{i=1}^{N} \langle J_i \rangle_P \xi_i\Big| + \lambda \frac{\sum_{\mu=1}^{\min(P,M)} (\sum_{j=1}^{N} \xi_j^{P+1-\mu} \xi_j)^2}{\min(M,P)} . \quad (28)$$

The second energy term is equal to the average squared overlap between $\boldsymbol{\xi}$ and an old pattern $\boldsymbol{\xi}^{\mu}$. The parameter λ controls the relative importance of the additional orthogonality constraints. In the present work we set $\lambda = 1$, and considering that there are N mutually orthogonal vectors in an N-dimensional space, we set $M = N - 1$. (We have not yet tried to optimize the values of λ and M.) Each new training pattern is then sampled by simulated annealing starting from an initial completely random pattern. The only difference is that the energy function in Eq. (17) now takes the form of Eq. (28).

The performance of the modified online learning algorithm is shown in Fig. 5. We find that error-free inference of the teacher's weight vector can be achieved with high probability after encountering $P \approx 1.9N$ training patterns. Compared with the results of Fig. 4, the additional orthogonality considerations indeed lead to a remarkable boost to the learning efficiency.

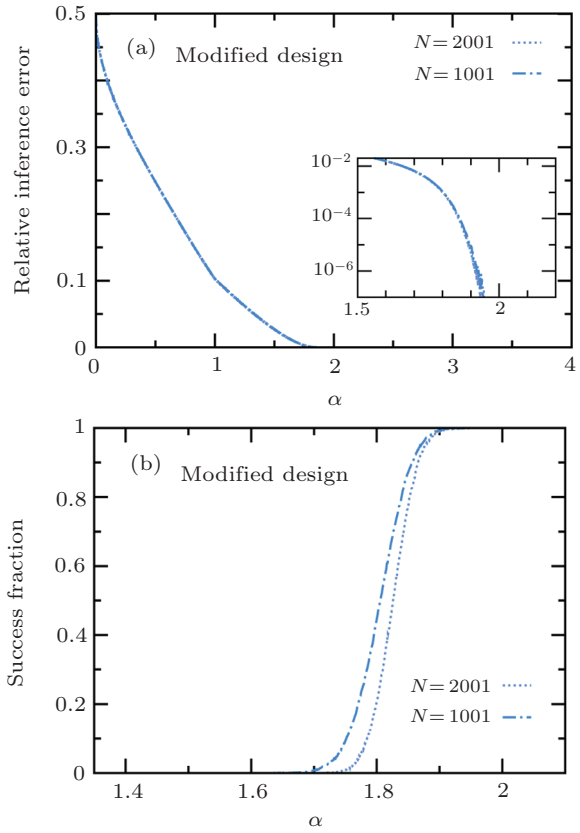

Fig. 5 Same as Fig. 3 and Fig. 4, but for active online learning under the designing constraint (15) and the additional orthogonality constraints, see Eq. (28). The number of stored training patterns is set to be $M = N$–1.

It may be possible to further improve the learning performance by optimizing the parameters λ and M of Eq. (28). Furthermore, the energy function (28) may not necessarily be the best way to incorporate both the designing constraint (15) and the additional orthogonality constraints. For example, it may be even better to consider all the P old patterns (instead of only the last M ones) with non-uniform weighting factors.

6 Discussion

In this work we considered the Bayesian active learning principle (6) to infer the teacher's weight vector of an N-dimensional Ising perceptron. Each new Ising training pattern is not randomly drawn as in passive learning but is designed with the aim of splitting the current version space into two equal sub-spaces. This designing principle was exactly implemented for small systems to achieve error-free inference using only N training samples (Fig. 2). When exhaustive enumeration becomes unfeasible for large systems, we derived a simple constraint (15) based on this principle and demonstrated that error-free inference is achievable with $P \approx 2.3N$ training samples (Fig. 4). The number of training samples was further reduced to $P \approx 1.9N$ after imposing additional orthogonality constraints on the training patterns (Fig. 5). On the other hand, the deductive reasoning algorithm discussed in this paper is guaranteed to achieve error-free inference with at most $N+\log_2 N$ queries and therefore is much superior to the Bayesian strategies discussed in Secs. 4 and 5.

In deriving the constraint Eq. (15) from the Bayesian principle (6), we have approximated the overlap probability profile $\mathcal{P}_P(q|\boldsymbol{\xi})$ (Eq. (11)) by a Gaussian distribution. Maybe this Gaussian assumption is only valid for sufficiently small values of the pattern density α. With the addition of training patterns, the volume of the version space becomes more or more small. At the same time the shape of the version space may become more and more irregular. One highly possible scenario is that, when α exceeds certain threshold value, the version space breaks up into many well-separated sub-spaces with each of them having a different set of mean weight values $\{\langle J_i\rangle_P\}$. As a consequence, the overlap probability profile $\mathcal{P}_P(q|\boldsymbol{\xi})$ should be described as a weighted sum of many distinct Gaussian distribution functions (one for each version sub-space), and then Eq. (15) should be modified accordingly.

From the academic point of view, active learning in the presence of ergodicity breaking is a very interesting challenge. With an accurate approximation to the overlap probability profile $\mathcal{P}_P(q|\boldsymbol{\xi})$, the efficiency of the active learning process may closely approach the theoretical limiting value of $\alpha = 1$. We hope that significant theoretical and algorithmic progresses will be made in the near future on this important research issue.

Acknowledgement

Numerical simulations were ccarrried out at the HPC cluster and Tianwen cluster of ITP-CAS. The author thanks Kim Sneppen for valuable discussions.

Appendix A: Derivation of Eq. (21)

The quantity A_i^+ as defined by Eq. (20a) is just the conditional probability of the random value x defined by $x = \sigma^{P+1}(\xi_i^{P+1} + \sum_{j\neq i}\xi_j^{P+1}J_j)$ being positive among all the weight vectors $\boldsymbol{J} \in \Sigma_P^{i+}$. The mean value of this random variable x is

$$\sigma^{P+1}(\xi_i^{P+1} + \sum_{j\neq i}\xi_j^{P+1}\langle J_j\rangle_P^{i+})\,, \tag{A1}$$

and its variance is

$$\sum_{j\neq i}\left(1-(\langle J_j\rangle_P^{i+})^2\right) + \sum_{j<k}{}' 2\xi_j^{P+1}\xi_k^{P+1}(\langle J_jJ_k\rangle_P^{i+} - \langle J_j\rangle_P^{i+}\langle J_k\rangle_P^{i+})\,, \tag{A2}$$

where $\sum'_{j<k}$ means the summation is over pair of indices $j,k \in \{1,2,\ldots,N\}$ satisfying $j<k$ and $j,k\neq i$. In the above two expressions, $\langle J_j\rangle_P^{i+}$ denotes the mean value of J_j in the version space Σ_P^{i+} and similarly for $\langle J_jJ_k\rangle_P^{i+}$.

To proceed, let us approximate $\langle J_j\rangle_P^{i+}$ simply by $\langle J_j\rangle_P$ and assume that $\langle J_jJ_k\rangle_P^{i+} = \langle J_j\rangle_P^{i+}\langle J_k\rangle_P^{i+}$. The variance (A2) then simplifies to be $\Delta(\boldsymbol{\xi}^{P+1}) - (1-\langle J_i\rangle_P^2)$ and after neglecting the correction term $(1-\langle J_i\rangle_P^2)$, is finally approximated to be $\Delta(\boldsymbol{\xi}^{P+1})$, where $\Delta(\boldsymbol{\xi}^{P+1})$ is computed according to Eq. (25). Under the assumption that the probability profile of the random variable x is well approximated by a Gaussian distribution, we obtain the following integration expression for the conditional probability A_i^+:

$$A_i^+ = \frac{1}{\sqrt{2\pi\Delta(\boldsymbol{\xi}^{P+1})}}\int_0^{+\infty}\exp\Big(-\frac{\big(x-\sigma^{P+1}\xi_i^{P+1}-\sigma^{P+1}\sum_{j\neq i}\xi_j^{P+1}\langle J_j\rangle_P\big)^2}{2\Delta(\boldsymbol{\xi}^{P+1})}\Big)\mathrm{d}x\,. \tag{A3}$$

An approximate expression for the conditional probability A_i^- as defined by Eq. (20b) can be derived by the same way:

$$A_i^- = \frac{1}{\sqrt{2\pi\Delta(\boldsymbol{\xi}^{P+1})}}\int_0^{+\infty}\exp\Big(-\frac{\big(x+\sigma^{P+1}\xi_i^{P+1}-\sigma^{P+1}\sum_{j\neq i}\xi_j^{P+1}\langle J_j\rangle_P\big)^2}{2\Delta(\boldsymbol{\xi}^{P+1})}\Big)\mathrm{d}x\,. \tag{A4}$$

The above two expressions are still not very convenient for numerical computations. By treating $\xi_i^{P+1}(1-\langle J_i\rangle_P)$ and $\xi_i^{P+1}(1+\langle J_i\rangle_P)$ as expansion small quantities with respect to the sum $\sum_{j=1}^N\xi_j^{P+1}\langle J_j\rangle_P$, we obtain that

$$A_i^+ = A + \xi_i^{P+1}(1-\langle J_i\rangle_P)\delta A\,, \qquad A_i^- = A - \xi_i^{P+1}(1-\langle J_i\rangle_P)\delta A\,, \tag{A5}$$

where

$$A = \frac{1}{\sqrt{2\pi\Delta(\boldsymbol{\xi}^{P+1})}}\int_0^{+\infty}\exp\Big(-\frac{\big(x-\sigma^{P+1}\sum_{j=1}^N\xi_j^{P+1}\langle J_j\rangle_P\big)^2}{2\Delta(\boldsymbol{\xi}^{P+1})}\Big)\mathrm{d}x\,,$$

$$\delta A = \frac{1}{\sqrt{2\pi\Delta(\boldsymbol{\xi}^{P+1})}}\exp\Big(-\frac{\big(\sum_{j=1}^N\xi_j^{P+1}\langle J_j\rangle_P\big)^2}{2\Delta(\boldsymbol{\xi}^{P+1})}\Big). \tag{6}$$

By inserting Eq. (A5) into Eq. (19) we obtain the experience accumulation formula (21).

References

[1] F. Rosenblatt, Psychological Review **65** (1958) 386.

[2] T. L. H. Watkin, A. Rau, and M. Biehl, Rev. Mod. Phys. **65** (1993) 499.

[3] A. Engel and C. Van den Broeck, *Statistical Mechanics of Learning*, Cambridge University Press, Cambridge, UK (2001).

[4] W. Krauth and M. Mézard, J. Phys. France **50** (1989) 3057.

[5] Y. Kabashima and S. Uda, Lect. Notes Artif. Intellig. **3244** (2004) 479.

[6] A. Braunstein and R. Zecchina, Phys. Rev. Lett. **96** (2006) 030201.

[7] H. Huang and Y. Kabashima, Phys. Rev. E **90** (2014) 052813.

[8] H. Huang, K. Y. M. Wong, and Y. Kabashima, J. Phys. A: Math. Theor. **46** (2013) 375002.

[9] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, Phys. Rev. Lett. **115** (2015) 128101.

[10] T. Obuchi and Y. Kabashima, J. Stat. Mech.: Theor. Exp. **2009** (2009) 12014.

[11] G. Györgyi, Phys. Rev. A **41** (1990) 7097.

[12] H. Sompolinsky, N. Tishby, and H. S. Seung, Phys. Rev. Lett. **65** (1990) 1683.

[13] N. Littlestone and M. K. Warmuth, *The Weighted Majority Game*, in *Proceedings of the 30th Annual Symposium on the Foundation of Computer Science*, IEEE, New York (1989) 256.

[14] M. Opper and D. Haussler, Phys. Rev. Lett. **66** (1991) 2677.

[15] H. S. Seung, M. Opper, and H. Sompolinsky, *Query by Committee*, in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, ACM, New York (1992) 287.

[16] M. Opper, Phys. Rev. Lett. **77** (1996) 4671.

[17] J. Feng, J. Phys. A: Math. Gen. **31** (1998) 4037.

[18] M. Rosen-Zvi, J. Phys. A: Math. Gen. **33** (2000) 7277.

[19] C. Baldassi, J. Stat. Phys. **136** (2009) 902.

[20] W. Kinzel and P. RuJán, Europhys. Lett. **13** (1990) 473.

[21] E. B. Baum, IEEE Trans. Neural Networks **2** (1991) 5.

[22] J. N. Hwang, J. J. Choi, S. Oh, and R. J. Marks II, IEEE Trans. Neural Networks **2** (1991) 131.

[23] O. Kinouchi and N. Caticha, J. Phys. A: Math. Gen. **25** (1992) 6243.

[24] H. S. Seung, H. Sompolinsky, and N. Tishby, Phys. Rev. A **45** (1992) 6056.

[25] T. L. H. Watkin and A. Rau, J. Phys. A: Math. Gen. **25** (1992) 113.

[26] Y. Kabashima and S. Shinomoto, *Acceleration of Learning in Binary Choice Problems*, in *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, ACM, New York (1993) 446.

[27] P. Sollich and D. Saad, *Learning from Queries for Maximum Information Gain in Imperfectly Learnable Problems* in G. Tesauro, D. S. Touretzky, and T. K. Leen, eds., *Advances in Neural Information Processing Systems*, **7** (1995) 287, MIT Press, Cambridge, MA.

[28] Y. Chen, X. Li, J. Liu, and Z. Ying, Applied Psychological Measurement **42** (2018) 24.

[29] G. E. P. Box, J. Amer. Stat. Assoc. **71** (1976) 791.

[30] H. Robbins, Bullet. Amer. Math. Soc. **58** (1952) 527.

[31] Y. LeCun, Y. Bengio, and G. E. Hinton, Nature (London) **521** (2015) 436.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA (2016).

[33] A. Huang, B. Sheldan, D. A. Sivak, and M. Thomson, arXiv:cond-mat.stat-mech/1805.07512, (2018).

[34] N. Rupprecht and D. C. Vural, arXiv:cond-mat.stat-mech/1810.06620, (2018).

[35] K. Ueltzhöffer, arXiv:q-bio.NC/1709.02341, (2017).

[36] K. J. Friston, M. Lin, C. D. Frith, *et al.,* Neural Computation **29** (2017) 2633.

[37] P. Boufounos and R. Baraniuk, *1-Bit Compressive Sensing* in *Proc. 42nd Annual Conference on Information Sciences and Systems*, IEEE (2008) 16.

[38] Y. Xu and Y. Kabashima, J. Stat. Mech.: Theor. Exp. **2013** (2013) 02041.

[39] J. M. Barzdin and R. V. Freivald, Soviet Mathematics Doklady **13** (1972) 1224.

[40] D. Angluin, Machine Learning **2** (1988) 319.

[41] N. Littlestone, Machine Learning **2** (1988) 285.

[42] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, Science **220** (1983) 671.

[43] M. Mézard, J. Phys. A: Math. Gen. **22** (1989) 2181.

[44] S. A. Solla and O. Winther, *Optimal Perceptron Learning: An Online Bayesian Approach* in D. Saad, ed., *On-Line Learning in Neural Networks*, Cambridge University Press, Cambridge, UK (1998) 379.

[45] V. Privman, ed., *Finite Size Scaling and Numerical Simulation of Statistical Systems*, World Scientific, Singapore (1990).

[46] T. Shinzato and Y. Kabashima, J. Phys. A: Math. Theor. **42** (2009) 015005.