Many-body approach to the dynamics of batch learning

K. Y. Michael Wong, S. Li, and Y. W. Tong
Department of Physics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
(Received 13 January 2000; revised manuscript received 20 May 2000)

Using the cavity method and diagrammatic methods, we model the dynamics of batch learning of restricted sets of examples, widely applicable to general learning cost functions, and fully taking into account the temporal correlations introduced by the recycling of the examples. The approach is illustrated using the Adaline rule learning teacher-generated or random examples.

PACS number(s): 87.10.+e, 87.18.Sn, 07.05.Mh, 05.20.−y

I. INTRODUCTION

An important problem in information processing is the extraction of the parameters underlying a set of examples, which is useful in such tasks as classification and regression [1]. This process, usually called *learning*, is often achieved by defining an appropriate energy function which reflects both the inability of the parameters in reproducing the training data, and the unlikelihood in fulfilling their prior expectation. The energy function is then minimized by a gradient descent process with respect to the parameters until a steady state is reached. For many years, physicists have gained success in studying the steady-state behavior of learning processes using equilibrium statistical mechanics [2]. On the other hand, the dynamics of learning was much less frequently addressed. The major difficulty is its complexity, since it typically involves the evolution of many microscopic parameters, each element affecting and being affected by others in a convolutional way. The challenge to the physicists is thus to describe this multivariate process using macroscopic variables.

Recently, much progress has been made on modeling the dynamics of *on-line* learning of infinite training sets [3–6]. In this model, an independent example is generated for each learning step. Since statistical correlations among the examples can be ignored, the dynamics can be simply described by instantaneous dynamical variables. This simplification results in a significant reduction in the complexity of analyzing learning dynamics, thereby leading to great advances in our understanding of on-line learning. In multilayer perceptrons, for instance, the persistence of a permutation symmetric stage which retards the learning process was well studied. Subsequent proposals to speed up learning were made, illustrating the usefulness of the on-line approach [6,7].

However, on-line learning represents an ideal case in which one has access to an almost infinite training set, whereas in many applications, the collection of training examples may be costly. In reality, the same restricted set of examples is recycled during the learning process. This introduces temporal correlations of the parameters in the learning history, which was the complexity circumvented in the models of on-line learning.

There have been attempts using statistical physics to describe the dynamics of learning of finite training sets. In batch learning, the same restricted set of examples is provided for *each* learning step. Using the dynamical mean-field theory, early work has been done on the steady-state behavior and asymptotic time scales in perceptrons with binary weights, rather than the continuous weights of more common interest [8]. Much benchmarking of batch learning has been done for linear learning rules such as Hebbian learning [9,10] or Adaline learning [11]. The work on Adaline learning was further extended to the study of linear perceptrons learning nonlinear rules [14,15]. However, not much work has been done on the learning of nonlinear rules with continuous weights. In this respect, it is interesting to note the recent attempts using the *dynamical replica theory* [9,10]. It approximates the temporal correlations during learning by instantaneous effective macroscopic variables. Further approximations facilitate results for nonlinear learning. Nevertheless, the rigor of these approximations remains to be confirmed in the general case.

In this paper, we model batch learning of restricted sets of examples, by considering the learning model as a many-body system. Each example makes a small contribution to the learning process, which can be described by linear-response terms in a sea of background examples. Two ingredients are important to our theory.

(a) *The cavity method*. Originally developed for magnetic systems and spin glasses [16], the method was adopted to learning in perceptrons [17], and subsequently extended to the teacher-student perceptron [18], the AND machine [19], the multiclass perceptron [20], the committee tree [21,22], Bayesian learning [23], and pruned perceptrons [24]. These studies only considered the equilibrium properties of learning, whereas here we are generalizing the method to study the dynamics [16]. It uses a self-consistency argument to compare the evolution of the activation of an example when it is absent or present in the training set. When absent, the activation of the example is called the *cavity activation*, in contrast to its generic counterpart when it is included in the training set.

(b) *The diagrammatic approach*. To describe the difference between the cavity activation and its generic counterpart of an example, we apply linear-response theory and use Green's function to describe how the influence of the added example propagates through the learning history. The Green's function is represented by a series of diagrams, whose averages over examples are performed by a set of pairing rules similar to those introduced for Adaline learning [11], as well as in the dynamics of layered networks [25].

Here we take a further step and use the diagrams to describe the changes from cavity to generic activations, as was done in [26], rather than the evolution of specific dynamical variables in the case of linear rules [11]. Hence our dynamical equations are widely applicable to any gradient-descent learning rule which minimizes an *arbitrary* cost function in terms of the activation. It fully takes into account the temporal correlations during learning, and is exact for large networks.

However, the solutions of the general dynamical equations are not tractable for nonlinear learning rules. In fact, the study of the general solution constitutes an area of future research and is beyond the scope of the present paper. Rather, for illustration, we apply the equations to a tractable case, namely Adaline learning, and extract results useful for efficient learning. Preliminary work has been presented recently [27].

The paper is organized as follows. In Sec. II we formulate the dynamics of batch learning. In Sec. III we introduce the cavity method and the dynamical equations for the macroscopic variables: (a) $G(t,s)$, the Green's function that propagates the response to a stimulus from time s to t; (b) $C(t,s)$, the correlation function, which is related to the Green's function via the fluctuation response relation; (c) $R(t)$, the overlap between the student and teacher perceptrons. In Sec. IV we describe the results useful for efficient learning applied to the case of the Adaline rule. The appendixes explain the diagrammatic approach in describing the Green's function, the fluctuation response relation, and the expressions for Adaline learning via the Laplace transform.

II. FORMULATION

Consider the single layer perceptron with $N \gg 1$ input nodes $\{\xi_j\}$ connecting to a single output node by the weights $\{J_j\}$ and often the bias θ as well. For convenience we assume that the inputs ξ_j are Gaussian variables with mean 0 and variance 1, and the output state is a function $f(x)$ of the *activation* x at the output node, where $x=\vec{J}\cdot\vec{\xi}+\theta$. For binary outputs, $f(x)=\mathrm{sgn}\,x$.

The network is assigned to ''learn'' $p \equiv \alpha N$ examples which map inputs $\{\xi_j^\mu\}$ to the outputs $\{S_\mu\}$ $(\mu=1,\ldots,p)$. In the case of random examples, S_μ are random binary variables, and the perceptron is used as a storage device. In the case of teacher-generated examples, S_μ are the outputs generated by a teacher perceptron with weights $\{B_j\}$ and often a bias ϕ as well, namely $S_\mu=f(y_\mu)$; $y_\mu=\vec{B}\cdot\vec{\xi}^\mu+\phi$.

Batch learning is achieved by adjusting the weights $\{J_j\}$ iteratively so that a certain cost function in terms of the activations $\{x_\mu\}$ and the output S_μ of all examples is minimized. Hence we consider a general cost function $E=-\Sigma_\mu g(x_\mu,y_\mu)$. The precise functional form of $g(x,y)$ depends on the adopted learning algorithm. In previous studies, $g(x,y)=-(S-x)^2/2$ with $S=\mathrm{sgn}\,y$ in Adaline learning [11–13], and $g(x,y)=xS$ in Hebbian learning [9,10].

To ensure that the perceptron fulfills the prior expectation of minimal complexity, it is customary to introduce a weight decay term. In the presence of noise, the gradient descent dynamics of the weights is given by

$$\frac{dJ_j(t)}{dt}=\frac{1}{N}\sum_\mu g'(x_\mu(t),y_\mu)\xi_j^\mu-\lambda J_j(t)+\eta_j(t), \quad (1)$$

where the prime represents partial differentiation with respect to x, λ is the weight decay strength, and $\eta_j(t)$ is the noise term at temperature T with

$$\langle \eta_j(t)\rangle=0 \quad \text{and} \quad \langle \eta_j(t)\,\eta_k(s)\rangle=\frac{2T}{N}\delta_{jk}\delta(t-s). \quad (2)$$

The dynamics of the bias θ is similar, except that no bias decay should be present according to consistency arguments [1],

$$\frac{d\theta(t)}{dt}=\frac{1}{N}\sum_\mu g'(x_\mu(t),y_\mu)+\eta_\theta(t). \quad (3)$$

III. THE CAVITY METHOD

Our theory is the dynamical version of the cavity method [16,21,22]. It uses a self-consistency argument to consider what happens when a new example is added to a training set. The central quantity in this method is the *cavity activation*, which is the activation of a new example for a perceptron trained without that example. Since the original network has no information about the new example, the cavity activation is random. Here we present the theory for $\theta=\phi=0$, skipping extensions to biased perceptrons. Denoting the new example by the label 0, its cavity activation at time t is $h_0(t)=\vec{J}(t)\cdot\vec{\xi}^0$. For large N, $h_0(t)$ is a Gaussian variable. Its covariance is given by the correlation function $C(t,s)$ of the weights at times t and s, that is, $\langle h_0(t)h_0(s)\rangle=\vec{J}(t)\cdot\vec{J}(s)\equiv C(t,s)$, where ξ_j^0 and ξ_k^0 are assumed to be independent for $j\neq k$. For teacher-generated examples, the distribution is further specified by the teacher-student correlation $R(t)$, given by $\langle h_0(t)y_0\rangle=\vec{J}(t)\cdot\vec{B}\equiv R(t)$.

Now suppose the perceptron incorporates the new example at the batch-mode learning step at time s. Then the activation of this new example at a subsequent time $t>s$ will no longer be a random variable. Furthermore, the activations of the original p examples at time t will also be adjusted from $\{x_\mu(t)\}$ to $\{x_\mu^0(t)\}$ because of the newcomer, which will in turn affect the evolution of the activation of example 0, giving rise to the so-called Onsager reaction effects. This makes the dynamics complex, but fortunately for large $p\sim N$, we can assume that the adjustment from $x_\mu(t)$ to $x_\mu^0(t)$ is small, and linear-response theory can be applied.

Suppose the weights of the original and new perceptron at time t are $\{J_j(t)\}$ and $\{J_j^0(t)\}$, respectively. Then a perturbation of Eq. (1) yields

$$\left(\frac{d}{dt}+\lambda\right)[J_j^0(t)-J_j(t)] = \frac{1}{N}g'(x_0(t),y_0)\xi_j^0+\frac{1}{N}\sum_{\mu k}\xi_j^\mu g''(x_\mu(t),y_\mu)\xi_k^\mu \times[J_k^0(t)-J_k(t)]. \quad (4)$$

The first term on the right-hand side describes the primary effects of adding example 0 to the training set, and is the driving term for the difference between the two perceptrons. The second term describes the many-body reactions due to the changes of the original examples caused by the added example, and is referred to as the Onsager reaction term. One should note the difference between the cavity and generic activations of the added example. The former is denoted by $h_0(t)$ and corresponds to the activation in the perceptron $\{J_j(t)\}$, whereas the latter, denoted by $x_0(t)$ and corresponding to the activation in the perceptron $\{J_j^0(t)\}$, is the one used in calculating the gradient in the driving term of Eq. (4). Since their notations are sufficiently distinct, we have omitted the superscript 0 in $x_0(t)$, which appears in the background examples $x_\mu^0(t)$.

The equation can be solved by the Green's-function technique, yielding

$$J_j^0(t)-J_j(t)=\sum_k \int ds\, G_{jk}(t,s)\left(\frac{1}{N}g_0'(s)\xi_k^0\right), \tag{5}$$

where $g_0'(s)\equiv g'(x_0(s),y_0)$ and $G_{jk}(t,s)$ is the *weight Green's function*, which describes how the effects of a perturbation propagates from weight J_k at learning time s to weight J_j at a subsequent time t. In the present context, the perturbation comes from the gradient term of example 0, such that integrating over the history and summing over all nodes give the resultant change from $J_j(t)$ to $J_j^0(t)$.

For large N the weight Green's function can be found by the diagrammatic approach explained in Appendix A. The result is self-averaging over the distribution of examples and is diagonal, i.e., $\lim_{N\to\infty}G_{jk}(t,s)=G(t,s)\delta_{jk}$, where

$$G(t,s)=G^{(0)}(t-s)+\alpha\int dt_1\int dt_2 G^{(0)}(t-t_1)\times\langle D_\mu(t_1,t_2)g_\mu''(t_2)\rangle G(t_2,s). \tag{6}$$

Here the bare Green's function $G^{(0)}(t-s)$ is given by

$$G^{(0)}(t-s)\equiv\Theta(t-s)\exp(-\lambda(t-s)), \tag{7}$$

and Θ is the step function. $D_\mu(t,s)$ is the *example Green's function* given by

$$D_\mu(t,s)=\delta(t-s)+\int dt' D_\mu(t,t')g_\mu''(t')G(t',s). \tag{8}$$

Our key to the macroscopic description of the learning dynamics is to relate the activation of the examples to their cavity counterparts, which is known to be Gaussian. Multiplying both sides of Eq. (5) by ξ_j^0 and summing over j, we have

$$x_0(t)-h_0(t)=\int ds\, G(t,s)g_0'(s). \tag{9}$$

In turn, the covariance of the cavity activation distribution is provided by the fluctuation-response relation explained in Appendix B,

$$C(t,s)=\alpha\int dt' G^{(0)}(t-t')\langle g_\mu'(t')x_\mu(s)\rangle + 2T\int dt' G^{(0)}(t-t')G(s,t'). \tag{10}$$

Furthermore, for teacher-generated examples, its mean is related to the teacher-student correlation given by

$$R(t)=\alpha\int dt' G^{(0)}(t-t')\langle g_\mu'(t')y_\mu\rangle. \tag{11}$$

To monitor the progress of learning, we are interested in three performance measures. (a) *Training error* ϵ_t, which is the probability of error for the training examples, and can be determined from the distribution $p(x|y)$ that the student activation of a trained example takes the value x for a given teacher activation y of the same example,

$$\epsilon_t=\int Dy\int dx\, p(x|y)\Theta(-xy). \tag{12}$$

(b) *Test error* ϵ_{test}, which is the probability of error when the inputs ξ_j^μ of the training examples are corrupted by an additive Gaussian noise of variance Δ^2. This is a relevant performance measure when the perceptron is applied to process data which are the corrupted versions of the training data. When $\Delta^2=0$, the test error reduces to the training error. Since the corrupted activation has an additional variance of $\Delta^2C(t,t)$, ϵ_{test} is given by

$$\epsilon_{\text{test}}=\int Dy\int dx\, p(x|y)H\left(\frac{x\,\text{sgn}\,y}{\sqrt{\Delta^2C(t,t)}}\right). \tag{13}$$

(c) *Generalization error* ϵ_g for teacher-generated examples, which is the probability of error for an arbitrary input ξ_j when the teacher and student outputs are compared. For an example with teacher activation y, the corresponding student activation is a Gaussian with mean $R(t)y$ and variance $C(t,t)$. Hence ϵ_g is given by

$$\epsilon_g=\frac{1}{\pi}\arccos\frac{R(t)}{\sqrt{C(t,t)}}. \tag{14}$$

The cavity method can be applied to the dynamics of learning with an arbitrary cost function. When it is applied to the Hebb rule, it yields results identical to [9]. Here for illustration, we present the results for the Adaline rule. This is a common learning rule and bears resemblance with the more common back-propagation rule. Theoretically, its dynamics is particularly convenient for analysis since $g''(x)=-1$, rendering the weight Green's function time translation invariant, i.e., $G(t,s)=G(t-s)$. In this case, the dynamics can be solved by the Laplace transform explained in Appendix C. Results useful for efficient learning are illustrated in the following section.

IV. RESULTS

(i) *Overtraining of* ϵ_g. As shown in Fig. 1, ϵ_g decreases at the initial stage of learning. However, for sufficiently weak

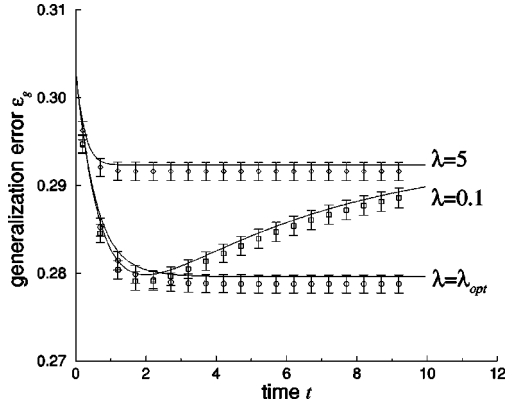

FIG. 1. The evolution of the generalization error at $\alpha=0.8$ and $T=0$ for different weight decay strengths λ. Theory: solid line, simulation: symbols.

weight decay, it attains a minimum at a finite learning time before reaching a higher steady-state value. This is called *overtraining* since at the later stage of learning, the perceptron is focusing too much on the specific details of the training set. In this case ϵ_g can be optimized by *early stopping*, i.e., terminating the learning process before it reaches the steady state. Similar behavior is observed in linear perceptrons [13–15].

This phenomenon can be controlled by tuning the weight decay λ. The physical picture is that the perceptron with minimum ϵ_g corresponds to a point with a magnitude $|\vec{J}^*|$. When λ is too strong, $|\vec{J}|$ never reaches this magnitude and ϵ_g saturates at a suboptimal value. On the other hand, when λ is too weak, $|\vec{J}|$ grows with learning time and is able to pass near the optimal point during its learning history. Hence the weight decay λ_{ot} for the onset of overtraining is closely related to the optimal weight decay λ_{opt} at which the steady state ϵ_g is minimum. Indeed, at $T=0$ and for all values of α, $\lambda_{ot}=\lambda_{opt}=\pi/2-1$; the coincidence of λ_{ot} and λ_{opt} is also observed previously [13]. This is illustrated in Fig. 2, which shows the early stopping time t_{es} at which ϵ_g is minimum in the learning history. For weak weight decay, t_{es} remains relatively insensitive to the value of λ, but diverges when λ approaches λ_{opt}.

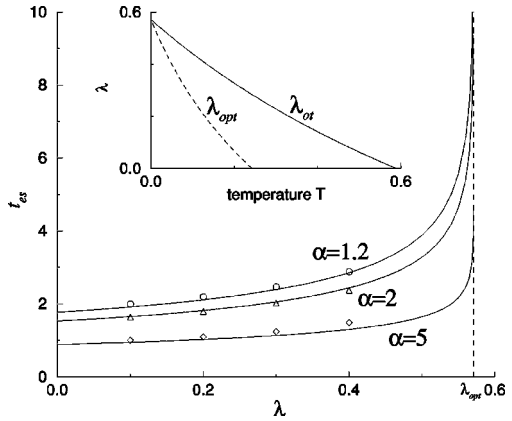

FIG. 2. Early stopping time t_{es} for different training set size α. Inset: The temperature dependence of the optimal weight decay λ_{opt} and the onset of overtraining λ_{ot} at $\alpha=5$.

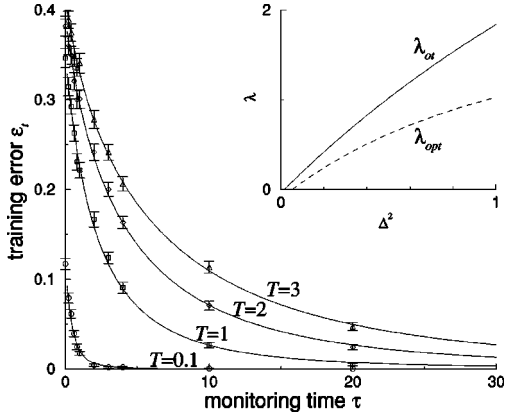

FIG. 3. The training error at $\alpha=0.1$ and $\lambda=5$ of the thermally averaged perceptron for random examples versus the duration τ of the monitoring period for thermal averaging. Inset: The lines of the optimal weight decay λ_{opt} and the onset of overtraining λ_{ot} of the test error for teacher-generated examples at $\alpha=3$ and $T=0.3$.

Early stopping for $\lambda<\lambda_{ot}=\lambda_{opt}$ can speed up the learning process, but cannot outperform the optimal result at the steady state. A recent empirical observation confirms that a careful control of the weight decay may be better than early stopping in optimizing generalization [28].

At nonzero temperatures, we find the new result that λ_{ot} and λ_{opt} may become different. Here we only mention the case of sufficiently large α. As shown in the inset of Fig. 2, λ_{opt} lies inside the region of overtraining, implying that even the best steady state ϵ_g is outperformed by some point during its own learning history. This means the optimal ϵ_g can only be attained by tuning *both* the weight decay and learning time. However, at least in the present case, computational results show that the improvement is marginal.

(ii) *Overtraining of* ϵ_{test}. Consider the effects of tuning the input noise from zero, when ϵ_{test} starts to increase from ϵ_t. At the steady state ϵ_t is optimized by $\lambda=0$ for $\alpha<1$, and by a relatively small $\lambda>0$ for $\alpha>1$. This means that ϵ_{test} is optimized with no or only little concern about the magnitude of J^2. However, when input noise is introduced, it adds a Gaussian noise of variance $\Delta^2 J^2$ to the activation distribution. The optimization of ϵ_{test} now involves minimizing the error of the training set without using an excessively large J^2. Thus the role of weight decay becomes important. Indeed, at $T=0$, $\lambda_{opt}=\alpha\Delta^2$ for random examples, whereas $\lambda_{opt}\propto\Delta^2$ approximately for teacher-generated examples. This illustrates how the environment in anticipated applications, i.e., the level of input noise, affects the optimal choice of perceptron parameters.

Analogous to the dynamics of ϵ_g, overtraining can occur when a sufficiently weak λ allows $\vec{J}$ to pass near the optimal point during its learning history. Indeed, at $T=0$ the onset of overtraining is given by $\lambda_{ot}=\lambda_{opt}$ for random examples, whereas $\lambda_{ot}\approx\lambda_{opt}$ for teacher-generated examples. At nonzero temperatures, λ_{ot} and λ_{opt} become increasingly distinct, and for sufficiently large α, $\lambda_{opt}<\lambda_{ot}$ as shown in the inset of Fig. 3, so that the optimal ϵ_{test} can only be attained by tuning *both* the weight decay and learning time.

(iii) *Average dynamics*. When learning has reached steady state, the dynamical variables fluctuate about their temporal averages because of thermal noises. If we consider a percep-

tron constructed using the thermally averaged weights $\langle J_j \rangle_{\rm th}$, we can then prove that it is equivalent to the perceptron obtained at $T=0$. This equivalence implies that for perceptrons with thermal noises, the training and generalization errors can be reduced by temporal averaging down to those at $T=0$.

We can further compute the performance improvement as a function of the duration τ of the monitoring period for thermal averaging, as confirmed by simulations in Fig. 3. In the asymptotic limit, the average overlap of the student J_j with the teacher B_j is independent of time, so that $\langle \vec{J} \rangle_{\rm th} \cdot \vec{B} = \lim_{t\to\infty} R(t)$. On the other hand, its amplitude is given by

$$\overline{C} = \lim_{T_0 \to 0} \frac{2}{\tau^2} \int_{T_0}^{T_0+\tau} dt_1 \int_{T_0}^{t_1} dt_2 C(t_1,t_2). \tag{15}$$

Using Eq. (C12), this reduces to

$$\overline{C} = \int \frac{dk}{k^2} \rho(k)(k-\lambda)\left[1+\frac{2}{\pi}(k-\lambda-1)\right] + T\int \frac{dk}{k}\rho(k)\left(\frac{e^{-k\tau}-1+k\tau}{k^2\tau^2/2}\right). \tag{16}$$

Note that the density of states $\rho(k)$ consists of a spectrum of relaxation modes $\exp(-kt)$ whose rate k lies in the range $k_{\rm min} \leqslant k \leqslant k_{\rm max}$, where $k_{\rm max}$ and $k_{\rm min}$ are $\lambda + (\sqrt{\alpha} \pm 1)^2$, respectively. For $\alpha < 1$, there is an additional relaxation mode with rate λ, which describes the relaxation by weight decay inside the $(N-p)$-dimensional solution space of zero training error. Hence the monitoring period scales as $k_{\rm min}^{-1}$ for $\alpha > 1$ and λ^{-1} for $\alpha < 1$. This thermal equilibration time agrees with the relaxation time proposed for asymptotic dynamics in [11].

We remark that the relaxation time for steady-state dynamics may not be the same as the convergence time for learning in the transient regime. For example, for vanishing α at $T=0$, a significant reduction of ϵ_t takes place in a time scale proportional to $(1+\lambda)^{-1}$. Hence for a vanishing weight decay, this time scale is independent of λ, which can be attributed to the dynamics being dominated by a growth of the projection onto the solution space of zero training error. On the other hand, the asymptotic relaxation time diverges as λ^{-1}, since the weight vector already resides in the solution space.

V. CONCLUSION

In summary, we have introduced a general framework for modeling the dynamics of learning based on the cavity method, which is applicable to general learning cost functions, though its tractable solutions are not generally available. It allows us to reach useful conclusions about overtraining and early stopping, input noise and temperature effects, transient dynamics, and average dynamics.

An example of extending the present study concerns the case of learning in biased perceptrons. Since no decay term is present in the learning of the bias in Eq. (3), its dynamics is modified. For sufficiently large weight decay, we find that there is an additional relaxation mode which may cause the bias to be learned slower than the weights. Details will be presented elsewhere.

We consider the present work as the beginning of an in-depth study of learning dynamics. An important issue is thus whether the analysis remains tractable for nonlinear learning rules. In general, $D_\mu(t,s)$ in Eq. (8) has to be expanded as a series. The dynamical equations are then the starting point of a perturbation theory. Another applicable area is the case of batch learning with very large learning steps, whose analysis remains simple due to its fast convergence [14]. Preliminary results are promising. The method can also be applied to on-line learning of restricted sets of examples.

An alternative general theory for learning dynamics is the dynamical replica theory [9]. It yields exact results for Hebbian learning, but for less trivial cases, the analysis is approximate and complicated by the need to solve replica saddle point equations at every learning instant. It is hoped that by adhering to an exact formalism, the cavity method can provide more fundamental insights when extended to multilayer networks.

ACKNOWLEDGMENTS

We thank A. C. C. Coolen and D. Saad for fruitful discussions. This work was supported by the Research Grant Council of Hong Kong (Grant Nos. HKUST6130/97P and HKUST615/99P).

APPENDIX A: THE GREEN'S FUNCTION

Substituting Eq. (5) into Eq. (4), we see that the Green's function satisfies

$$\left(\frac{d}{dt}+\lambda\right) G_{jk}(t,s) = \delta(t-s)\,\delta_{jk} + \frac{1}{N}\sum_{\mu i} \xi_j^\mu g_\mu''(t) \xi_i^\mu G_{ik}(t,s). \tag{A1}$$

Introducing the bare Green's function $G^{(0)}(t-s)$ in Eq. (7),

$$G_{jk}(t,s) = G^{(0)}(t-s)\,\delta_{jk} + \frac{1}{N}\sum_{\mu i} \int dt' G^{(0)}(t-t') \times \xi_j^\mu g_\mu''(t') \xi_i^\mu G_{ik}(t',s). \tag{A2}$$

This equation is represented diagrammatically in Fig. 4(a). We use a slanted line to represent an example bit, the top and bottom ends of the line corresponding to the example label and node label, respectively. A filled circle represents $g_\mu''(t)$. Thin and thick lines represent the bare and dressed Green's functions $G^{(0)}(t-s)$ and $G(t,s)$, respectively. The iterative solution to Eq. (A2) can be represented by the series of diagrams in Fig. 4(b). It is convenient to concurrently introduce the *example* Green's function $D_\mu(t,s)$ as shown in Fig. 4(c).

The average over the distribution of example inputs is done by pairing of example or node labels and is represented by dashed lines connecting the vertices above or below the solid lines. Pairing of example and node labels yields factors of 1 and α, respectively. Noting that crossing diagrams do not contribute [11], the two Green's functions can be expressed in terms of the self-energies Σ and Π_μ, via the Dyson's equations in Fig. 4(d). The self-energies are defined in Fig. 4(e), and are characterized by having the first node or example paired with the last one only. The self-energies can

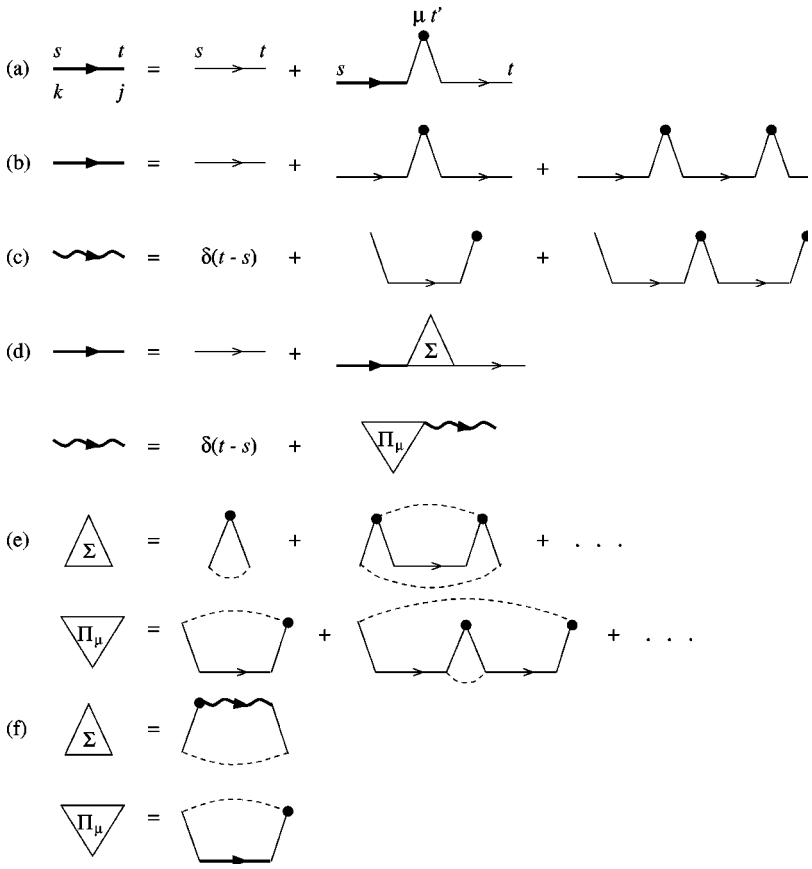

FIG. 4. (a) Diagrammatic representation of Eq. (A2); (b) iterative solution to Eq. (A2); (c) the example Green's function; (d) Dyson's equations; (e) the self-energies; (f) the self-energies in terms of the Green's functions.

in turn be expressed in terms of the Green's functions as in Fig. 4(f), thus allowing for self-consistent solutions.

After eliminating the self-energies, the results of the diagrammatic analysis are given by Eqs. (6) and (8).

APPENDIX B: THE FLUCTUATION RESPONSE RELATION

In terms of the bare Green's function, the solution to the dynamical equation (1) is

$$J_j(t)=\frac{1}{N}\sum_\mu \int dt' G^{(0)}(t-t')g'_\mu(t')\xi_j^\mu + \int dt' G^{(0)}(t-t')\,\eta_j(t'). \tag{B1}$$

Multiplying both sides by $J_j(s)$ and summing over j, we have

$$C(t,s)=\alpha\int dt' G^{(0)}(t-t')\langle g'_\mu(t')x_\mu(s)\rangle + \int dt' G^{(0)}(t-t')\sum_j J_j(s)\,\eta_j(t'). \tag{B2}$$

The correlation between $J_j(s)$ and $\eta_j(t')$ can be considered by comparing the learning process with another one which is noiseless between $t'-\epsilon$ and $t'+\epsilon$, but is otherwise identical. Denoting the weight of this alternative process by $J_j^{\setminus \eta(t')}$, we have

$$J_j(s)=J_j^{\setminus \eta(t')}(s)+\int_{t'-\epsilon}^{t'+\epsilon} dt'' G(s,t'')\,\eta_j(t''). \tag{B3}$$

Noting that $J_j^{\setminus \eta(t')}(s)$ is uncorrelated with $\eta_j(t')$, and $\eta_j(t'')$ has a δ-function correlation with $\eta_j(t')$, we arrive at Eq. (10).

Similarly, multiplying both sides by B_j and summing over j, we arrive at Eq. (11).

APPENDIX C: ADALINE SOLUTION BY LAPLACE TRANSFORM

For Adaline learning with $g'(x,y)=\mathrm{sgn}\,y-x$, the Laplace transforms to Eqs. (6)–(9), respectively, are

$$\tilde{G}(z)=\frac{1}{z+\lambda}-\alpha\tilde{D}(z)\tilde{G}(z), \tag{C1}$$

$$\tilde{D}(z)=1-\tilde{D}(z)\tilde{G}(z), \tag{C2}$$

$$\tilde{x}_0(z)-\tilde{h}_0(z)=\tilde{G}(z)\left[\frac{\mathrm{sgn}\,y_0}{z}-\tilde{x}_0(z)\right]. \tag{C3}$$

Solving for $\tilde{G}(z)$ in Eqs. (C1) and (C2), and applying the inverse Laplace transform, the Green's function becomes

$$G(t)=\int dk\,\rho(k)e^{-kt}, \tag{C4}$$

where $\rho(k)$ is the density of states,

$$\rho(k)=\Theta(1-\alpha)\,\delta(k-\lambda)+\frac{\sqrt{(k_{\max}-k)(k-k_{\min})}}{2\pi(k-\lambda)}, \tag{C5}$$

with $k_{\max},k_{\min}=\lambda+(\sqrt{\alpha}\pm 1)^2$, respectively. This enables us to solve for $x_0(t)$ in terms of $h_0(t)$ in Eq. (C3),

$$x_0(t)=\frac{\operatorname{sgn} y_0}{\alpha}\int dk\,\rho(k)(k-\lambda)\left(\frac{1-e^{-kt}}{k}\right)+\int dt' K(t,t')h_0(t'), \tag{C6}$$

where

$$K(t,t')=\delta(t-t')-\frac{\Theta(t-t')}{\alpha}\int dk\,\rho(k)(k-\lambda)e^{-k(t-t')}. \tag{C7}$$

The Laplace transform to the teacher-student correlation in Eq. (11) is

$$\tilde{R}(z)=\frac{\alpha}{z+\lambda}\langle \tilde{g}'_\mu(z)y_\mu\rangle. \tag{C8}$$

Using the properties that $\langle y_\mu \operatorname{sgn} y_\mu\rangle=\sqrt{2/\pi}$ and $\langle \tilde{h}_\mu(z)y_\mu\rangle=\tilde{R}(z)$, we have

$$\tilde{R}(z)=\sqrt{\frac{2}{\pi}}\frac{\alpha}{z}\frac{\tilde{G}(z)}{1+\tilde{G}(z)}, \tag{C9}$$

whose inverse Laplace transform yields

$$R(t)=\sqrt{\frac{2}{\pi}}\int dk\,\rho(k)(k-\lambda)\left(\frac{1-e^{-kt}}{k}\right). \tag{C10}$$

The Laplace transform to the fluctuation response relation Eq. (10) is

$$\tilde{C}(z,w)=\frac{1}{z+\lambda}\left[\alpha\langle \tilde{g}'_\mu(z)\tilde{x}_\mu(w)\rangle+\frac{2T}{z+w}\tilde{G}(w)\right]. \tag{C11}$$

Noting further that $\langle \tilde{h}_\mu(z)\tilde{h}_\mu(w)\rangle=\tilde{C}(z,w)$, it can be cast into a symmetric form whose inverse Laplace transform yields

$$C(t,s)=\int dk\,\rho(k)(k-\lambda)\left[1+\frac{2}{\pi}(k-\lambda-1)\right]\left(\frac{1-e^{-kt}}{k}\right)\times\left(\frac{1-e^{-ks}}{k}\right)+T\int dk\,\rho(k)\left(\frac{e^{-k|t-s|}-e^{-k(t+s)}}{k}\right). \tag{C12}$$

For the activation distribution $p(x|y)$, we consider Eq. (C6) and the Gaussian distribution of the cavity activation, which implies that $x_0(t)$ is a Gaussian with mean and variance,

$$\langle x_0(t)\rangle=\frac{\operatorname{sgn} y_0}{\alpha}\int dk\,\rho(k)(k-\lambda)\left(\frac{1-e^{-kt}}{k}\right)+\int dt' K(t,t')R(t'), \tag{C13}$$

$$\langle x_0(t)^2\rangle-\langle x_0(t)\rangle^2=\int dt_1\int dt_2 K(t,t_1)K(t,t_2)[C(t_1,t_2)-R(t_1)R(t_2)]. \tag{C14}$$

For random examples, the evolution of the parameters is the same, except that $R(t)$ is identically zero, and the term in the square brackets in Eq. (C12) is replaced by 1.

[1] C. M. Bishop, *Neural Networks for Pattern Recognition* (Clarendon Press, Oxford, 1995).
[2] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation* (Addison Wesley, Redwood City, CA, 1991).
[3] M. Biehl and H. Schwarze, Europhys. Lett. **20**, 733 (1992).
[4] D. Saad and S. Solia, Phys. Rev. Lett. **74**, 4337 (1995).
[5] D. Saad and S. Solla, Phys. Rev. E **52**, 4225 (1995).
[6] D. Saad and M. Rattray, Phys. Rev. Lett. **79**, 2578 (1997).
[7] M. Rattray, D. Saad, and S. Amari, Phys. Rev. Lett. **81**, 5461 (1998).
[8] H. Horner, Z. Phys. B: Condens. Matter **86**, 291 (1992); **87**, 371 (1992).
[9] A. C. C. Coolen and D. Saad, in *Advances in Neural Information Processing Systems*, edited by M. S. Kearns, S. A. Solla, and D. A. Cohn (MIT Press, Cambridge, MA, 1999), Vol. 11.
[10] H. C. Rae, P. Sollich, and A. C. C. Coolen, in *Advances in Neural Information Processing Systems* (Ref. [9]).
[11] J. Hertz, A. Krogh, and G. I. Thorbergssen, J. Phys. A **22**, 2133 (1989).
[12] M. Opper, Europhys. Lett. **8**, 389 (1989).
[13] A. Krogh and J. A. Hertz, J. Phys. A **25**, 1135 (1992).
[14] S. Bös and M. Opper, J. Phys. A **31**, 4835 (1998).
[15] S. Bös, Phys. Rev. E **58**, 833 (1998).
[16] M. Mézard, G. Parisi, and M. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore, 1987).
[17] M. Mézard, J. Phys. A **22**, 2181 (1989).
[18] M. Bouten, J. Schietse, and C. van den Broeck, Phys. Rev. E **52**, 1958 (1995).
[19] M. Griniasty, Phys. Rev. E **47**, 4496 (1993).
[20] F. Gerl and U. Krey, J. Phys. A **28**, 6501 (1995).
[21] K. Y. M. Wong, Europhys. Lett. **30**, 245 (1995).
[22] K. Y. M. Wong, in *Advances in Neural Information Processing Systems*, edited by M. C. Mozer, M. I. Jordan, and T. Petsche (MIT Press, Cambridge, MA, 1997), Vol. 9.
[23] M. Opper and O. Winther, Phys. Rev. Lett. **76**, 1964 (1996).
[24] K. Y. M. Wong, in *Theoretical Aspects of Neural Computation*, edited by K. Y. M. Wong, I. King, and D. Y. Yeung (Springer, Singapore, 1998).
[25] K. Y. M. Wong, C. Campbell, and D. Sherrington, J. Phys. A **28**, 1602 (1995).
[26] K. Y. M. Wong, Europhys. Lett. **38**, 631 (1997).
[27] S. Li and K. Y. M. Wong, in *Advances in Neural Information Processing Systems*, edited by S. A. Solla, T. K. Leen, and K.-R. Müller (MIT Press, Cambridge, MA, 2000), Vol. 12.
[28] L. K. Hansen, J. Larsen, and T. Fog, IEEE Int. Conf. Acoustics, Speech, Signal Processing **4**, 3205 (1997).