In *Online Learning in Neural Networks*, The Newton Institute Series, D. Saad (ed.), Cambridge University Press, Cambridge, 1998.

Exact and Perturbation Solutions for the Ensemble Dynamics

Todd K. Leen

Dept. of Computer Science and Engineering and
Dept. of Electrical and Computer Engineering
Oregon Graduate Institute of Science & Technology P.O. Box 91000
Portland, Oregon
tleen@cse.ogi.edu

Abstract

This paper presents two approaches to characterize the dynamics of weight space probability density starting from a master equation. In the first, we provide a class of algorithms for which an exact evaluation of the integrals in the master equation is possible. This enables the time evolution of the density to be calculated at each time step *without approximation*. In the second, we expand earlier work on the small noise expansion to a complete perturbation framework. As an example application, we give a perturbative solution for the equilibrium density for the LMS algorithm. Finally, we use the perturbation framework to review annealed learning with schedules of the form $\mu(t) = \mu_0/t^p$.

1 Introduction

Several of the contributions to this volume apply order parameter techniques to describe the dynamics of on-line learning. In these approaches, the description assumes the form of a set of ordinary differential equations that describe the motion of macroscopic quantities, the order parameters, from which observables such as generalization error can be computed directly, without intermediate averaging. The (typically non-linear) differential equations are usually solved numerically to provide the dynamics throughout the training. Recently, this author and colleagues have obtained analytic results from the order parameter equations applied to the late time (asymptotic) convergence of annealed learning [1]. However, most of the application of these techniques has involved numerical integration. Indeed, the ability to track the ensemble dynamics at all times by solution of ordinary differential equations establishes an attractive analysis tool.

In contrast, this paper examines analytic approaches to the master equation that describes the dynamics of the weight space probability density in time. These dynamics are expressed as partial differential equations, of infinite order, that do not ordinarily admit simple numerical solution. Instead, one identifies useful approximations and uses the equations in the applicable domain. These approaches have proved most valuable to describe learning with very small learning rates and for asymptotic phenomena such as equilibrium distributions [2, 3] and the late-time behavior of annealed learning [4], though there has been some application to describe transients as well [5, 6, 7].

Here we develop this approach in two ways. First, we identify a class of algorithms, and a particular case of interest, for which the integrals appearing in the master equation can be solved in closed form. This provides an *exact*, closed form expression for the operator that generates the time displacements of the system. This does not imply that the density is available analytically at each time step, but rather that it can be calculated (numerically perhaps) *without approximation* by simple matrix multiplication. The solution for the density as a function of time does not rely on numerical integration of any kind, and is given without approximation.

Secondly, we develop a perturbation expansion for solutions of the master equation. Here the weight space density is expressed directly as a power series in the learning rate. At each power (order of perturbation theory), one solves a set of partial differential equations to obtain the density to that order. This framework includes, as a special case, earlier work that treated only the lowest order contribution without consideration of how they fit into a perturbation scheme. We also adopt the perturbation scheme to treat annealed learning with rates of the form $\mu_0/t^p, \;\; 0 < p \leq 1$. Retaining the leading terms, we recover the classic asymptotic normality results on $1/t$ annealing, and analogous, though markedly different, results $p \neq 1$.

2 Ensemble Dynamics

Our starting point is learning rules of the form

$$w(n+1) \;=\; w(n) \;+\; \mu(n)\, Q(w(n), x(n)) \tag{2.1}$$

where $w(n)$ and $\mu(n)$ are the weight estimate and learning rate (gain) at time n, $x(n)$ is the datum input to the algorithm at time n, and $Q(w, x)$ is the parameter update function. We assume that the input, which may be an input/target pair in the case of supervised learning, is drawn i.i.d from the distribution $p(x)$[1]

In stochastic *gradient descent* algorithms, the update function $Q(w(n), x)$ is minus the gradient of the instantaneous cost $E(w, x)$. The ensemble average of the instantaneous cost is the *true* cost $E(w)$. The latter drives the corresponding deterministic, or batch-mode, gradient descent algorithm.

The learning rate μ may be independent of time, or it may follow a specified time-dependence, or may change in time in response to the progress of the learning [9, 10, 11]. Constant learning rates are commonly employed during the initial phases (and sometimes through all phases) of stochastic learning algorithms. Constant μ allows the system to converge on local optima at rates comparable to the equivalent batch algorithm (e.g. exponential convergence in quadratic minima). However, most problems require that the learning rate be annealed at late times in order to obtain convergence (in mean square, or with probability one) to a local optima. The learning rate may be either scalar, or matrix. Though the former is simpler and far more commonly implemented, optimal convergence rates require consideration of *matrix* learning rates as for example in [4, 12, 13].

Our goal is to describe the describe the ensemble dynamics of w in terms of an evolving probability distribution $P(w, n)$ on the weights. Towards this end, we develop a master equation for $P(w, n)$ from the learning rule (2.1). The first step is to recognize the that single-step transition probability, conditioned on the datum x is a Dirac delta function whose arguments satisfy the learning rule (2.1)

$$T(w|w', x, n) \;=\; \delta(\, w - w' - \mu(n)\, Q(w', x)\,) \;\; . \tag{2.2}$$

We recover the net transition probability by integrating the conditional transition probability over the measure on the input data[2] x

$$\begin{aligned} T(w|w', n) \;&=\; \int \rho(x)\, T(w|w', x, n)\, dx \\ &\equiv\; \langle\, \delta\,(\, w - w' - \mu(n) Q(w', x)\,)\,\rangle_x \;\; . \end{aligned} \tag{2.3}$$

Finally, the transition probability provides a *Master equation* that describes the dynamics of the density on w

$$\begin{aligned} P(w, n+1) \;&=\; \int dw' \;\; T(w|w', n)\; P(w', n) \\ &=\; \int dw'\, dx\, \rho(x)\; \delta(w - w' - \mu(n) Q(w', x))\; P(w', n) \end{aligned} \tag{2.4}$$

[1] The interesting case of correlated inputs is discussed in [8].

[2] We assume that the data x are sampled i.i.d. from the density $\rho(x)$. Experimentally, this corresponds to random sampling *with replacement* from a training dataset.

This provides a complete description of the ensemble dynamics of the learning rule (2.1) including: generalization, equilibrium distributions, convergence with annealed learning $\mu(n)$, and escape from local optima. Unfortunately, for most learning rules of interest, we cannot evaluate the integral on the right hand side in closed form. Thus approximation schemes are usually required.

We proceed in two ways. First we will give a family of algorithms, and a particular example of interest, for which the integral can be complete in closed form, and hence the time evolution specified exactly. Since the integral moves the probability density forward one step in time, it can be regarded as the operator that generates time displacements. Hence we refer to this as an exact integration of the time evolution operator. Following the exact solution, we will discuss approximations to the integral in (2.4) coupled with perturbation solutions for $P(w, n)$ that give approximate analytic solutions.

3 Exact Integration of the Time Evolution Operator

A class of algorithms for which the transition probability (2.3) assumes a simple, closed form can be constructed with reasonable constraints. Although the constraints may seem unnatural at first, a bit later we will give a concrete example that was suggested independently in a very natural way.

Suppose that the learning rate μ is constant in time. Further, suppose that the update function $Q(w', x)$ is such that for any w', $Q(w', x)$ is *piecewise constant in* x, and can only assume a *finite number* of possible values. We denote the values that Q assumes by q_i, $i = 1 \ldots m$ and the sets on which Q attains these values by $S_i(w')$

$$S_i(w') \;=\; \{\, x \,|\, Q(w', x) = q_i \,\} \quad . \tag{3.1}$$

We denote the data measure associated with each of these sets by f_i

$$f_i(w') \;\equiv\; \int_{S_i(w')} \rho(x)\, dx \quad . \tag{3.2}$$

With this construction, the transition probability (2.3) reduces to a weighted sum of delta functions

$$T(w \,|\, w') \;=\; \sum_{i=1}^{m} f_i(w')\, \delta(w - w' - \mu\, q_i) \tag{3.3}$$

and the right-hand side of the master equation (2.4) integrates simply to

$$P(w, n+1) \;=\; \sum_{i=1}^{m} P(w - \mu\, q_i, n)\; f_i(w - \mu\, q_i) \quad . \tag{3.4}$$

Equation (3.4) generates the dynamics of the $P(w, n)$ without approximation. It provides, for example, a means to numerically compute the evolution of the density *exactly*, provided numerical values of the measures f_i are available. The expression tells us to compute the density at w at time step $n + 1$ by locating the points $w - \mu\, q_i$ that can jump to the point w in one time step, and accumulate the densities at those points, each weighted by the f_i. The latter is just the probability that one chooses an x that generates the required jump q_i.

3.1 An Example: Sign-of-the-Gradient Descent

Most commonly used algorithms are some variant of stochastic gradient descent; for which the update $Q(w, x)$ is minus the gradient of an instantaneous cost function

$$Q(w, x) \;=\; -\nabla_w E(w, x) \quad .$$

If instead, we base the update on the *sign* of the gradient, we retrieve a simple algorithm that is a member of our class of algorithms with integrable time-evolution operator. There are two variants, one of which we briefly describe here. A more thorough development is in [14].

The algorithm proceeds as follows. At each time step, select a single weight at random (with equal probability of selecting any weight) and update it according to the sign of the instantaneous gradient

$$Q_j(w,x) \;=\; -\xi_j \;\mathrm{sign}\left[\frac{\partial E(w,x)}{\partial w_j}\right] \tag{3.5}$$

where $\xi_j \in \{0,1\}$ are indicator variables that denote which weight is chosen for updating.
For this system, $q_i = 0, \pm 1$. For a system of N weights, one finds

$$\begin{aligned} P(w,n+1) - P(w,n) \;=\; & -\frac{1}{2\,N}\sum_{j=1}^{N}\Big[D_j^{(1)}(w+\mu_j)\,P(w+\mu_j,n) \\ & \qquad -D_j^{(1)}(w-\mu_j)\,P(w-\mu_j,n)\Big] \\ & +\frac{1}{2\,N}\sum_{j=1}^{N}\Big[D_{jj}^{(2)}(w+\mu_j)\,P(w+\,u_j,n) \\ & \qquad -2\,D_{jj}^{(2)}(w)\,P(w,n) \\ & \qquad +\,D_{jj}^{(2)}(w-\mu_j)\,P(w-\mu_j,n)\Big] \quad . \end{aligned} \tag{3.6}$$

Here we have rewritten the terms in f_i in terms of the drift vector

$$D_j^{(1)}(w) = N\,\langle\,\xi_j\,Q_j(w,x)\,\rangle_{x,\xi}$$

and diagonal terms of the diffusion matrix

$$D_{jk}^{(2)}(w) = N\,\langle\,\xi_j\xi_k\,Q_j(w,x)\,Q_k(w,x)\,\rangle_{x,\xi}$$

whose relation to the f_i for this problem are given in [14], and can be easily derived from (3.5). The quantity μ_j in (3.6) is a displacement of length μ along the j^{th} weight coordinate.
The system (3.6) is of the form of a finite difference approximation to a Fokker-Planck equation. However, the equation describes the dynamics to all orders, without approximation. There are several interesting features. With proper initialization, the dynamics can be developed by simple matrix multiplication. Suppose that the initial density $P(w,0)$ is non-zero *only* at weight values corresponding to integer multiples of μ_j, $w = (i_1\,\mu_1,\; i_2\,\mu_2,\; \ldots)^T$. Then, since at each iteration weights only change by $\pm\mu_j$ or zero, the density $P(w,n)$ at any time step n will have support only at weight values that are integer multiples of μ_j. Thus, the density is confined to a *rectangular grid* with spacings μ_j along the w_j axis. The dynamics is generated simply by matrix multiplication. One forms a vector $P(n)$ whose entries consist of the densities $P(w,n)$ at the grid points, and a matrix A whose entries contain the coefficients of the density in equation (3.6). The evolution of the ensemble density is then given by

$$P(n+1) \;=\; (1\;+\;A)\,P(n) \;=\; (1+A)^{n+1}\;P(0) \quad .$$

The evolution matrix $1+A$ is sparse, since the learning rule only connects nearest neighbor points on the weight grid. For a one-dimensional configuration space, the evolution matrix $1+A$ is tri-diagonal, and the computation is particularly easy to set up and execute. Equilibrium distributions correspond to the null space of A, and first passage time calculations reduce to the solution of a linear system.
Also of interest are problems with trapping states; values of w for which $Q(w_*,x) = \nabla_w E(w_*,x) = 0$ for *all possible* x. One example is regression problems with noiseless targets generated by a function that can be exactly fit by the network. In general, there are several such states that we denote by $w_*^{[i]}$, $i = 1,\ldots,p$. It is straightforward to verify that the master equation (2.4) has equilibria consisting of Dirac delta functions at the $w_*^{[i]}$

$$P_{eq}(w) \;=\; \sum_{i=1}^{p} a_i\delta(w - w_*^{[i]})$$

where the occupation probabilities a_i for each solution weight are determined by the initial distribution $P(w,0)$.

Under standard stochastic gradient descent, the system will converge into the set of optima $w_*^{[i]}$. For learning driven by the sign of the gradient, these solutions are only accessible if they all fall on the weight grid with spacings μ_j, and if all the initial density also lies on this grid. If this is *not* the case, then instead of convergence to these solution weights, the density will, at late times, execute oscillations between the grid states neighboring the $w_*^{[i]}$.

Finally, we point out that in most problems, such trapping states are *not* present. In order to achieve convergence of the weights in the usual stochastic gradient descent, one must anneal the learning rate. In sign of the gradient learning, annealing corresponds to shrinking the grid size.

4 Perturbation Approach

In physics, it is common to approach a problem for which a closed solution cannot be found by treating it as a perturbation of a similar problem that one *can* solve in closed form. The attempt to analytically describe the ensemble dynamics of on-line learning algorithms fits this paradigm well.

Gardiner [15] describes a small noise perturbation approach for diffusion (Fokker-Planck) systems. Our development here is similar in form, but appropriately modified for the master equation we encounter in stochastic learning. The material here extends the basic small-noise expansion introduced by Heskes [3] in two ways. First, we emphasize that the combination of Heskes' small noise expansion with a perturbation expansion for $P(w,n)$ is a natural structure that incorporates the now-familiar Gaussian solutions as the lowest-order term in a series expansion. This series is closely related to the Edgeworth and Gram-Charlier expansions of classical statistics [16]. Secondly, we extend the treatment in a minor way to accommodate fixed-schedule learning rate annealing.

Our starting point is the continuous time Kramers-Moyal [15, for example] expansion of the master equation (2.4). This is obtained by expanding the transition probability (2.3) in a Taylor series in the learning rate μ. We also change from integer to continuous time, but gloss over the intricacies of the transition, referring the interested reader to [3]) Furthermore, as appropriate to most annealed learning algorithms, we restrict the form of $\mu(t)$ to a simple power law $\mu(t) = \mu_0/t^p$. One finds (in one dimension for notational simplicity)

$$\partial_t P(w,t) \;=\; \sum_{k=1}^{\infty} \frac{(-1)^k}{k!} \left(\frac{\mu_0}{t^p}\right)^k \;\partial_w^k \left(\left\langle Q^k(w,x) \right\rangle_x \; P(w,n) \right) \quad , \tag{4.1}$$

where ∂_t denotes the partial derivative with respect to time, and ∂_w^k denotes the k^{th} partial derivatives with respect to w. Next we decompose the trajectory $w(t)$ into a deterministic piece and a stochastic piece

$$w \;\equiv\; \phi(t) \;+\; \mu_0^\gamma \, f(t) \, \xi \qquad \text{or} \qquad \xi \;=\; \left(\frac{1}{\mu_0^\gamma \, f(t)}\right) \, (w - \phi(t)) \tag{4.2}$$

where $\phi(t)$ is the deterministic trajectory and ξ are the fluctuations. Apart from the factor $\mu_0^\gamma \, f(t)$ that scales the fluctuations, this is identical to Heskes' formulation for constant learning in [3]. We will obtain the proper value for the unspecified exponent γ, and the form of the function $f(t)$ from homogeneity requirements.

Next, the dependence of the jump moments $\left\langle Q^i(w,x) \right\rangle_x$ on μ_0 is explicated by a Taylor series expansion about the deterministic path ϕ. The coefficients in this series expansion are denoted

$$\alpha_i^{(j)} \;\equiv\; \left. \frac{\partial^j \left\langle Q^i(w,x) \right\rangle_x}{\partial w^j} \right|_{w=\phi} \quad .$$

For convenience, we define a new time variable

$$s \;=\; t$$

and transform the differential operators and densities in (4.1) as dictated by (4.2)

$$\begin{aligned} \partial_t &= \partial_s - \frac{1}{\mu_0^\gamma f(s)} \frac{d\phi(s)}{ds} \partial_\xi - \left(\frac{f'}{f}\right) \xi \partial_\xi \\ \partial_w &= \frac{1}{\mu_0^\gamma f(s)} \partial_\xi \\ P(w,t) &= (\mu_0^\gamma f(s))^{-1} \Pi(\xi, s) \ , \end{aligned} \tag{4.3}$$

where $\Pi(\xi, s)$ is the density of the fluctuations.

Finally, we rewrite (4.1) in terms of ϕ and ξ and the expansion of the jump moments, using the transformations (4.3), and suitably resumming the series. These transformations leave equations of motion for the deterministic trajectory $\phi(s)$ and the density of the fluctuations

$$\frac{d\phi}{ds} = \left(\frac{\mu_0}{s^p}\right) \alpha_1^{(0)}(\phi) = \left(\frac{\mu_0}{s^p}\right) \langle Q(\phi, x) \rangle_x \tag{4.4}$$

$$\begin{aligned} \partial_s \Pi &= \left(\frac{f'}{f}\right) \partial_\xi (\xi \Pi) \\ &+ \sum_{m=2}^{\infty} \sum_{i=1}^{m} \frac{(-1)^i}{i!(m-i)!} \alpha_i^{(m-i)} \frac{{\mu_0}^{i(1-2\gamma)+m\gamma}}{s^{ip}} f(s)^{m-2i} \partial_\xi^i (\xi^{(m-i)} \Pi) \ . \end{aligned} \tag{4.5}$$

This last is the basic form that we use in developing the perturbation expansion.

We are going to use the system (4.4) and (4.5) to describe the late-time evolution of learning systems with either small, constant learning rate, or with annealed learning rate. In either case, we will concentrate on the behavior near asymptotically stable fixed points w_* of the vector field $\langle Q(w,x) \rangle_x$. Thus we are assuming that the system has evolved long enough so that $\phi \to w_*$ and we can evaluate the $\alpha_i^{(j)}$ *at* the local optimum. We assume that we have $\alpha_1^{(1)} < 0$ in a neighborhood of w_*, i.e. that the linearization is non-zero[3]. For a gradient descent algorithm, this corresponds to a *quadratic minimum* w_*, with positive-definite cost function curvature

$$H = -\nabla_w \nabla_w E(w_*) = -\alpha_1^{(1)}(w_*) \ .$$

With this structure assumed for the vector field in the vicinity of the fixed point w_*, we can proceed to specify the constant γ and the form of $f(s)$. To insure that for each m, the terms in the sum over i in (4.5) are homogeneous in powers of μ_0, we take

$$\gamma = 1/2 \ . \tag{4.6}$$

Similarly, to insure that for each value of m, the terms in the sum over i are homogeneous in time, we take

$$f(s) = \frac{1}{s^{p/2}} \ . \tag{4.7}$$

4.1 Perturbation Expansion for Constant Learning Rate

For constant learning rate, $p = 0$, we rescale time as $\mu_0 s \equiv t$ and rewrite the equation of motion for the fluctuations as

$$\partial_t \Pi = \left(L_0 + \mu_0^{1/2} L_1 + \mu_0^1 L_2 + \ldots\right) \Pi(\xi, t) = \sum_{i=0}^{\infty} \mu^{i/2} L_i \Pi(\xi, t) \tag{4.8}$$

[3] In the multi-dimensional case, all of the eigenvalues of the matrix $\alpha_1^{(1)}$ should have negative real part.

where the action of the operators L_i are defined by the expansion (4.5) with the choice for γ (4.6 and $f(s)$ (4.7), and the time re-scaling above. The action of the first several operators are

$$L_0 F(\xi) \equiv -\partial_\xi(\alpha_1^{(1)}\,\xi\,F) + \frac{1}{2}\alpha_2^{(0)}\partial_\xi^2 F \tag{4.9}$$

$$L_1 F(\xi) \equiv -\frac{1}{2}\alpha_1^{(2)}\,\partial_\xi(\xi^2 F) + \frac{1}{2}\alpha_2^{(1)}\partial_\xi^2(\xi F) - \frac{1}{3!}\alpha_3^{(0)}\partial_\xi^3 F \tag{4.10}$$

$$\vdots$$

In the limit $\mu_0 \to 0$ only the L_0 term of (4.8) contributes. By evaluating the drift $\alpha_1^{(1)}$ and diffusion $\alpha_2^{(0)}$ at a local optimum w_*, (4.8) describes the time evolution of the fluctuation density about the local optimum, in the limit of small learning rate. Solving this lowest order piece for the equilibrium density gives the now-familiar Gaussian approximation to the asymptotic density. However, this is only a piece of the picture.

The lowest-order solution can be augmented by a perturbation expansion for the density. We write

$$\Pi(\xi,t) \equiv \Pi^{(0)} + \mu^{1/2}\Pi^{(1)} + \mu^1\Pi^{(2)} + \mu^{3/2}\Pi^{(3)} + \ldots \tag{4.11}$$

where the $\Pi^{(i)}$ are functions to be solved for. Next, we substitute this expansion of Π into the equation of motion (4.8) and equate the coefficients of like powers of μ on the left and right-hand sides. This leaves the set of perturbation equations

$$\partial_t\Pi^{(0)} = L_0\,\Pi^{(0)} \tag{4.12}$$

$$\partial_t\Pi^{(1)} = L_0\Pi^{(1)} + L_1\Pi^{(0)} \tag{4.13}$$

$$\partial_t\Pi^{(2)} = L_0\Pi^{(2)} + L_1\Pi^{(1)} + L_2\Pi^{(0)} \tag{4.14}$$

$$\vdots$$

The solution strategy is to solve (4.12) for $\Pi^{(0)}$, use this solution in (4.13) and solve the latter for $\Pi^{(1)}$, then use these two solutions to solve (4.14) for $\Pi^{(2)}$ etc. Thus, we solve the equations order-by-order in perturbation, obtaining approximations to Π in powers of $\mu^{1/2}$. Below we briefly describe how this is accomplished.

4.1.1 Solving the Perturbation Equations

Solution of the system (4.12), (4.13), etc relies on obtaining a complete set of eigenfunctions of the operator L_0

$$L_0\,f_n = \lambda_n\,f_n$$

and it's conjugate

$$L_0^\dagger\,g_n = \lambda_n\,g_n \ .$$

These functions form a bi-orthogonal set

$$\int g_k(\xi)\,f_j(\xi)\,d\xi \equiv = (g_k, f_j) = \delta_{ij} \ . \tag{4.15}$$

The operator L_0 corresponds to an Ornstein-Uhlenbeck process and has well-known eigenfunctions [15] (recall our constraint $\alpha_1^{(1)} < 0$)

$$f_n(\xi) = \sqrt{\frac{|\alpha_1^{(1)}|}{\pi\alpha_2^{(0)}}}\ \exp\left(-\frac{|\alpha_1^{(1)}|}{\alpha_2^{(0)}}\,\xi^2\right)\ g_n(\xi)$$

$$g_n(\xi) = \frac{1}{\sqrt{2^n n!}}\,H_n\left(\sqrt{\frac{|\alpha_1^{(1)}|}{\alpha_2^{(0)}}}\,\xi\right)$$

$$\lambda_n = n\,|\alpha_1^{(1)}|, \qquad n = 0,1,2,\ldots \tag{4.16}$$

with H_n the n^{th} order Hermite polynomials

It is trivial to verify that the solution of the lowest order equation (4.12) is

$$\Pi^{(0)}(\xi,t) \;=\; \sum_{n=0}^{\infty} a_n \exp(-\lambda_n t) f_n(\xi) \quad ,$$

where the coefficients a_n are determined by the initial distribution. Hence as $t \to \infty$, $\Pi^{(0)}$ converges to $f_0(\xi)$ which is a zero mean Gaussian equilibrium distribution with variance $\sigma_\xi^2 = \alpha_2^{(0)}/(2|\alpha_1^{(1)}|)$. Hence the lowest order approximation to the equilibrium distribution on w is a Gaussian peaked up about the local optimum w_* with variance $\sigma_w^2 = \mu\alpha_2^{(0)}/(2|\alpha_1^{(1)}|)$.

The higher order corrections to the equilibrium density can be calculated using (4.13), (4.14) etc. with the left-hand sides set to zero. We expand each perturbation correction $\Pi^{(i)}$ in the basis of eigenfunctions f_n

$$\Pi^{(i)}(\xi) \;\equiv\; \sum_{j=0}^{\infty} \gamma_j^{(i)}\, f_j(\xi) \quad .$$

Thus, the perturbation expansion develops each $\Pi^{(i)}$, and thus the total solution Π, as a series of Hermite polynomials times Gaussians (the eigenfunctions f_n). In this respect, our perturbation expansion recalls the classical Edgeworth and Gram-Charlier expansions for density functions [16].

The coefficients $\gamma_j^{(i)}$ are obtained by substituting this assumed form of the solution into the appropriate equation of motion for the $\Pi^{(i)}$, and using the bi-orthogonality relation (4.15). For example, the first-order correction satisfies

$$L_0\Pi^{(1)} \;=\; L_0 \sum_{i=1}^{\infty} \gamma_j^{(1)}\, f_j \;=\; -L_1\, \Pi^{(0)} \quad ,$$

or

$$\sum_{i=1}^{\infty} \gamma_j^{(1)}\, \lambda_j\, f_j \;=\; -L_1\, \Pi^{(0)} \quad .$$

Taking the inner product with g_k, we obtain

$$\gamma_k^{(1)} = -\frac{1}{\lambda_k}\, \left(g_k,\, L_1\Pi^{(0)}\right), \quad k \neq 0 \quad . \tag{4.17}$$

So the first order perturbation correction for the equilibrium density is

$$\Pi^{(1)} = \sum_{k=1}^{\infty} \frac{-1}{\lambda_k} \left(g_k,\, L_1\Pi^{(0)}\right)\, f_k(\xi) \quad . \tag{4.18}$$

Using this solution along with the equation for the second order correction (4.14), we obtain

$$\begin{aligned} \Pi^{(2)} \;&=\; \sum_{i=1}^{\infty} \frac{-1}{\lambda_i} \left[(g_i,\, L_1\Pi^{(1)}) + (g_i,\, L_2\Pi^{(0)})\right]\, f_i(\xi) \\ &=\; \sum_{i,j=1}^{\infty} \frac{(g_i,\, L_1 f_j)(g_j,\, L_1\Pi^{(0)})}{\lambda_i\, \lambda_j}\; f_i(\xi) \;-\; \sum_{i=1}^{\infty} \frac{(g_i,\, L_2\Pi^{(0)})}{\lambda_i}\, f_i(\xi) \quad . \end{aligned} \tag{4.19}$$

Clearly, one can calculate the corrections to arbitrary order provided one can evaluate the appropriate matrix elements $(g_i, L_j f_k)$ of the perturbation operators. Given the corrections, the density is re-assembled according to (4.11).

4.1.2 LMS Equilibrium Density

To illustrate, we give examples of perturbation solutions for the equilibrium density $P_e(w)$ of a single weight, linear neuron trained by the LMS algorithm (stochastic gradient descent on the mean squared error). The target signals are generated by a linear teacher neuron with weight w_* and zero mean Gaussian noise with variance σ^2_{noise} added to the output. The inputs were zero mean Gaussian with variance σ^2_x. The details of the calculations, and the results were originally presented in [17]. The equilibrium density is given in terms of the weight error $v \equiv w - w_*$. One finds for the first few perturbative solutions

$$\begin{aligned} P_e^{(0)}(v) &= \frac{1}{\sqrt{\pi\mu\sigma^2_{noise}}} \exp(-v^2/\mu\sigma^2_{noise}) \\ P_e^{(1)}(v) &= \frac{3}{4}\sigma_x^2 \left(-1 + \frac{2v^2}{\mu\sigma^2_{noise}}\right) P_e^{(0)}(v) \\ P_e^{(2)}(v) &= \frac{9}{32}\sigma_x^4 \left(-1 - \frac{4v^2}{\mu\sigma^2_{noise}} + \frac{4v^4}{\mu^2\sigma^4_{noise}}\right) P^{(0)} \quad . \end{aligned} \tag{4.20}$$

For comparison, we also include the equilibrium density obtained from the nonlinear Fokker-Planck equation obtained by truncating the Kramers-Moyal expansion (4.1) at second order in μ. This Fokker-Planck equation has a closed form equilibrium solution [18]

$$P_e^{FP}(v) = \frac{1}{B\left(1/2, 1/2 + 1/(3\mu\sigma_x^2)\right)} \left(1 + \frac{3\sigma_x^2}{\sigma^2_{noise}} v^2\right)^{-(1+\frac{1}{3\mu\sigma_x^2})} \tag{4.21}$$

where $B(\cdot,\cdot)$ is the Riemann beta function.

The solid curves in Figures 1 and 2 show the perturbation solution for the equilibrium densities obtained with input variance $\sigma_x^2 = 4$ and noise variance $\sigma^2_{noise} = 1$ for learning rates $\mu = 0.05$ and $\mu = 0.1$ respectively. Also shown are empirical estimates of the densities obtained from Monte Carlo simulations (dashed curves). The density predicted by the Fokker-Planck equation (dotted curve from equation (4.21)) is more peaked than the empirical density.

For the smaller of the two learning rates depicted in figure 1, the first order perturbation solution (Fig. 1 b) fits the empirical density quite well, and offers a substantial improvement relative to the zero order (Gaussian) approximation (Fig. 1 a). At higher learning rate, as in figure 2, the first order perturbation solution fits less well, and adding the next non-zero order term $\Pi^{(4)}$ further degrades the fit. The reader should note that $\mu = 0.1$ is already quite high as, for this input variance level, the equilibrium density has finite covariance only for $\mu < 0.1666$. The failure of the perturbation expansion at relatively high learning rates is not unexpected; perturbation expansions in physics are often asymptotic rather than convergent, and this is likely to be the case here. Finally, we note that another approach to specifying the LMS equilibrium density is given by [19].

The perturbation technique, while discussed above for a one-dimensional problem, applies in principle to problems of arbitrary dimension. To carry out the program, one needs to be able to find eigenfunctions of the multidimensional operator L_0

$$L_0 f(\xi) = \nabla_\xi \cdot \left(-\alpha_1^{(1)} \xi f(\xi)\right) + \frac{1}{2}\nabla_\xi \cdot \left(\alpha_2^{(0)} \nabla_\xi f(\xi)\right) = \lambda f(\xi) \quad .$$

This will pose a technical difficulty unless one can find coordinates in which this partial differential equation is separable. Since the drift $\alpha_1^{(1)}$ and diffusion $\alpha_2^{(0)}$ matrices may not be diagonalized by the same transformation – this may indeed be difficult.

There is, however, a class of problems for which these two matrices are simultaneously diagonalized by the same coordinate transformation. Fluctuations about the global optimum in supervised learning problems where the network is capable of representing the true regression function (so-called "realizable" problems) are treatable. In this case, using the negative log-likelihood of the data as a cost function, one finds that the

drift and diffusion matrices are scalar multiples of one-another. In this case, the eigenfunction equations separate in coordinates that diagonalize the curvature matrix H.

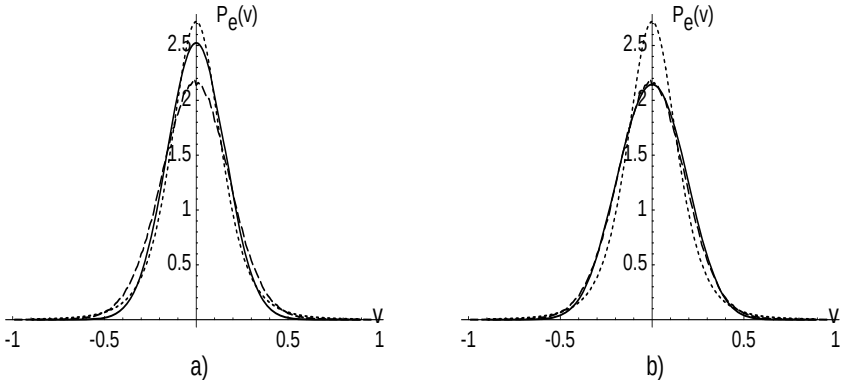

Figure 1: Comparision of equilibrium densities for 1-D LMS with $\mu = .05$, $\sigma_{noise} = 1$, $\sigma_x^2 = 4$. The dashed curves are the density estimated from an ensemble of simulations. The dotted curves are the density predicted from the Fokker-Planck equation obtained by truncating the Kramer-Moyal expansion (4.1). The solid curves are those obtained from the perturbation expansion. a) Using the 0^{th} order approximation $\Pi^{(0)}$. b) Using terms through the 1st order correction.

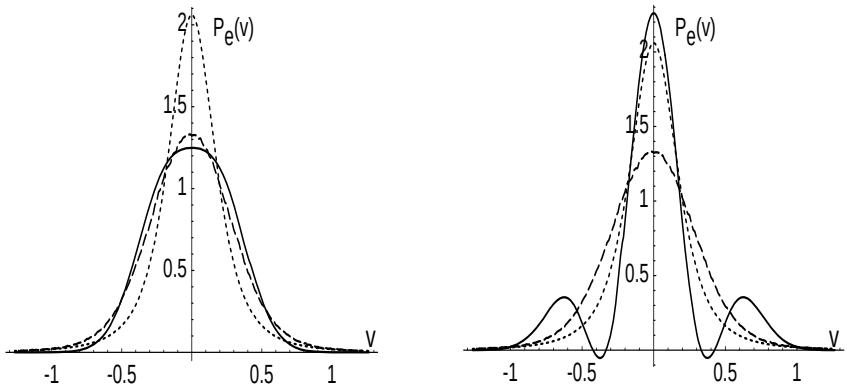

Figure 2: Same as figure 1, but with $\mu = 0.1$. a) Using terms through the first order correction $\Pi^{(1)}$. b) Using terms through the fourth order correction $\Pi^{(4)}$.

4.2 Annealed Learning

As discussed earlier, most practical applications of on-line learning employ some form of learning rate annealing. Power-law decay of the learning rate can be analyzed in the present framework in a straightforward manner. We recall our earlier assumption that the learning rate behaves as

$$\mu(s) \;=\; \frac{\mu_0}{s^p}$$

with $0 < p \leq 1$. We re-write the equations of motion for the fluctuations (4.5), using the chosen form of γ (4.6) and the form of $f(s)$ in (4.7), and writing now, the general n-dimensional form

$$\begin{aligned} \partial_s \, \Pi(\xi) \quad &= \quad \nabla_\xi \cdot \left[\left(-\frac{\mu_0}{s^p} \, \alpha_1^{(1)} \; - \; \frac{p}{2s} \right) \, \xi \, \Pi \right] \\ &\quad + \frac{\mu_0}{2s^p} \, \nabla_\xi \cdot \left(\alpha_2^{(0)} \, \nabla_\xi \Pi \right) \\ &\quad + \mathcal{O}\left(\frac{\mu_0}{s^p} \right)^{3/2} \end{aligned} \tag{4.22}$$

where $\alpha_1^{(1)}$ is the *negative definite* drift matrix, and $\alpha_2^{(0)}$ is the positive definite diffusion matrix.

Since $0 < p \leq 1$, at late times the right-hand side of (4.22) is dominated by the terms explicitly written explicitly. Since we are primarily concerned with the asymptotic dynamics, it is adequate to retain only the given terms. Precisely which terms dominate depends on the value of p, i.e. on the rate at which the learning rate is annealed. We will briefly review the results for classical annealing ($p = 1$) and for slower rates. These results are compared with an order parameter approach in [1].

Classical Annealing

For $p = 1$, the explicit terms on the right-hand side of (4.22) all decay as $1/s$. Relative to the constant learning rate case, the substantive change in the fluctuation dynamics is the modification of the drift by the term $-1/2s$ to form an *effective drift*. We confine attention to gradient descent algorithms for which $-\alpha_1^{(1)} = H$ is the cost function curvature. We assume that this is positive definite in a neighborhood of a local optimum w_* where we carry out the analysis[4]. Then the effective drift is

$$d_{eff} \;=\; \left(\frac{\mu_0}{s}\,H \;-\; \frac{1}{2s}\right)\,\xi \;\equiv\; \frac{1}{s}\,A\,\xi$$

In order to get a stable system, and hence a normalizable equilibrium density, the matrix A must be positive definite. This clearly requires that

$$\mu_0 \;>\; \frac{1}{2\,\lambda_{min}} \tag{4.23}$$

where λ_{min} is the smallest eigenvalue of the curvature H.

We next transform (4.22) to the new time coordinate $b = \ln s$ and recover the equation of motion valid at large times

$$\partial_b\,\Pi(\xi,t) \;=\; \nabla_\xi\cdot(A\,\xi\,\Pi) \;+\; \frac{\mu_0}{2}\,\nabla_\xi\cdot\left(\alpha_2^{(0)}\nabla_\xi\Pi\right) \quad . \tag{4.24}$$

Assuming that the criticality condition (4.23) is met, this gives rise to a zero-mean Gaussian equilibrium density for the fluctuations ξ, or using (4.2) with $\phi(t) = w_*$, a Gaussian equilibrium density for $\sqrt{s}\,(w - w_*)$. Thus we recover the classical result [20, 21] ; provided the criticality condition (4.23) is met, the random variable $\sqrt{s}\,v \equiv \sqrt{s}\,(w - w_*)$ is *asymptotically normal*.

This asymptotic normality implies a convergence rate for the algorithm. Given a finite covariance Σ_ξ for the fluctuations, the covariance of the weight error $v \equiv w - w_*$, denoted Σ_v, is inversely proportional to time

$$\Sigma_v \;=\; \frac{\mu_0}{s}\,\Sigma_\xi \quad .$$

and consequently the expected squared weight error drops off inversely with time

$$E\left[\,|v|^2\,\right] \;=\; \text{Trace}\;(\Sigma_v) \;=\; \frac{\mu_0}{s}\;\text{Trace}\,(\Sigma_\xi)$$

provided the criticality condition (4.23) is satisfied. This appears to be the optimal convergence rate that can be sustained over an extended period of time [22].

If the criticality condition is *not* met, the convergence will be *slower* than $1/s$. One can derive the asymptotic convergence rate by developing from (4.24) equations of motion for the second moments $R_\xi \;\equiv\; E[\xi_i\,\xi_j]$. Then using their solution, compute

$$E[|v|^2] \;=\; \text{Trace}(E[vv^T]) = \frac{\mu_0}{s}\text{Trace}(R_\xi) \quad .$$

[4] We are assuming as before that the system has evolved until the deterministic piece of the trajectory $\phi(s)$ is arbitrarily close to a local optimum.

The solution (derived by another approach in [4]) is

$$E[|v|^2] = \sum_{k=1} n\tilde{C}_{kk}(s_0)\left(\frac{s_0}{s}\right)^{2\mu_0\lambda_k} + \frac{\mu_0^2\,\tilde{\alpha^{(0)}}_{2\,kk}}{(2\mu_0\lambda_k - 1)}\left[\frac{1}{s} - \frac{1}{s_0}\left(\frac{s_0}{s}\right)^{2\mu_0\lambda_k}\right] \tag{4.25}$$

where λ_k are the eigenvalues of the curvature H (at the local optimum w_*), and $\tilde{C}_{kk}$ and $\tilde{\alpha^{(0)}}_{2\,kk}$ are the diagonal elements of the weight error covariance, and the diffusion matrix, in coordinates for which H is *diagonal*.

Equation (4.25) shows that when the criticality condition is met, one has $1/s$ decay of the expected squared weight error, but when the criticality condition is *not* met, the convergence is as $(1/s)^{2\mu_0\lambda_{min}}$, i.e. *slower* than $1/s$.

Since the criticality condition (4.23) relates the minimal required μ_0 to the *unknown* cost function curvature, one is not guaranteed to achieve the optimal convergence rate for an arbitrarily chosen μ_0. Since the actual convergence rate given in (4.25) can be *much* slower than optimal, this situation has led many researchers [9, 10, 4, 12, 11] to devise algorithms that attempt to adaptively set μ_0.

Alternative Annealing Schedules

The situation is rather different for $p < 1$. Referring to (4.22), at late times the right-hand side is dominated by the terms in $1/s^p$, and the $1/s$ and $\mathcal{O}(1/s^p)^{3/2}$ terms can be neglected. Then there is *no* criticality condition, and the fluctuations are asymptotically normal regardless of μ_0. By arguments similar to those in the last section, this implies that the expected squared weight error drops off asymptotically as

$$E[|v|^2] \propto \frac{1}{s^p} \tag{4.26}$$

This is, since $p < 1$, *slower* than the optimal rate achievable for $1/s$ annealing. However, there is *no* critical value of μ_0 to reach the $1/s^p$ rate[5].

5 Summary

We discussed the dynamics of on-line learning from the perspective of the master equation for the probability density on the weights. We found a class of algorithms, a member of which proceeds by updating the weight according to the *sign* of the gradient (sometimes called Manhattan learning), for which the time-evolution can be expressed in closed form. For such algorithms, the analysis proceeds without approximation, and without the need to integrate a set of differential equations. The existence of an exactly integrable model, even for a rather severely restricted set of algorithms, provides a potentially useful analysis tool.

We also developed a perturbation approach, to analyze the density of fluctuations about a deterministic trajectory. This approach is most useful to characterize learning in the vicinity of a local optimum. Our application to equilibrium distributions for constant learning, showed both the efficacy, and the limits of the technique. The perturbation equations are also applicable to annealed learning. The discussion here concentrated only on the asymptotic behavior, which requires retaining only the lowest order non-trivial terms. However it seems likely that some information on the late-time transient behavior can be obtained by retaining further orders in perturbation.

Finally, we note that all the examples discussed within the perturbation framework assumed that the curvature matrix $-\alpha_1^{(1)}$ at the local optimum is positive definite. If in some direction the curvature is

[5] This may seem rather paradoxical, as one could take p arbitrarily close to 1 and thereby achieve convergence rate arbitrarily close to $1/s$ regardless of μ_0. However, the analysis here only describes what happens when one reaches the asymptotic equilibrium distribution on ξ. The time to relax to that equilibrium distribution is *not* addressed by this analysis

zero, then the minimum is quartic (or of higher order) along that direction. One can address quartic minima in one dimension in a straightforward way, by suitable choice of the scaling exponent γ. More interesting, and yet to be addressed, is the case of singular curvature in multiple dimensions where some directions are quadratic, and some quartic or higher order. Such cases presumably require theoretical development analogous to center manifold theory in bifurcating deterministic dynamical systems.

Acknowledgements

The author would like to thank the Newton Institute and the International Human Frontier Science Program for travel support, and the NSF for continued research support under grant ECS-9704094.

References

[1] Todd K. Leen, Bernhard Schottky, and David Saad. Two approaches to optimal annealing. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems 10*. The MIT Press, 1998.

[2] Tom M. Heskes and Bert Kappen. Learning processes in neural networks. *Physical Review A*, 44:2718–2726, 1991.

[3] Tom M. Heskes. *Learning Processes in Neural Networks*. PhD thesis, Department of Medical Physics and Biophysics, University of Nijmegen, The Netherlands, June 1993.

[4] Todd K. Leen and Genevieve B. Orr. Optimal stochastic search and adaptive momentum. In J.D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, San Francisco, CA., 1994. Morgan Kaufmann Publishers.

[5] G. Radons, H.G. Schuster, and D. Werner. Fokker-Planck description of learning in backpropagation networks. In *International Neural Network Conference - INNC 90, Paris*, pages II 993–996. Kluwer Academic Publishers, July 1990.

[6] Tom M. Heskes, Eddy T.P. Slijpen, and Bert Kappen. Learning in neural networks with local minima. *Physical Review A*, 46(8):5221–5231, 1992.

[7] Genevieve B. Orr and Todd K. Leen. Weight space probability densities in stochastic learning: II. Transients and basin hopping times. In Giles, Hanson, and Cowan, editors, *Advances in Neural Information Processing Systems, vol. 5*, San Mateo, CA, 1993. Morgan Kaufmann.

[8] W. Wiegerinckand T. Heskes. On-line learning with time-correlated patterns. *Europhysics Letters*, 28:451–455, 1994.

[9] J. H. Venter. An extension of the robbins-monro procedure. *Annals of Mathematical Statistics*, 38:117–127, 1967.

[10] Christian Darken and John Moody. Towards faster stochastic gradient search. In J.E. Moody, S.J. Hanson, and R.P. Lipmann, editors, *Advances in Neural Information Processing Systems 4*. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[11] Noboru Murata, Klaus-Robert Muller, Andreas Ziehe, and Shun ichi Amari. Adaptive on-line learning in changing environments. In *Advances in Neural Information Processing Systms 9*. The MIT Press, 1997.

[12] Genevieve B. Orr and Todd K. Leen. Using curvature information for fast stochastic search. In *Advances in Neural Information Processing Systems 9*. The MIT Press, 1997.

[13] Howard H. Yang and Shun ichi Amari. The efficiency and the robustness of natural gradient descent learning rule. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems, 10*. The MIT Press, 1998.

[14] Todd K. Leen and John E. Moody. Stochastic manhattan learning, an exact time-evolution operator for the ensemble dynamics. *Physical Review E*, 56:1262–1265, 1997.

[15] C.W. Gardiner. *Handbook of Stochastic Methods, 2nd Ed.* Springer-Verlag, Berlin, 1990.

[16] Sir Maurice Kendall and Alan Stuart. *The Advanced Theory of Statistics, Volume 1 Distribution Theory*. MacMillan Publishing Co., New York, fourth edition, 1977.

[17] Genevieve B. Orr. *Dynamics and Algorithms for Stochastic Search*. PhD thesis, Oregon Graduate Institute, October 1996.

[18] Todd K. Leen and John E. Moody. Weight space probability densities in stochastic learning: I. Dynamics and equilibria. In Giles, Hanson, and Cowan, editors, *Advances in Neural Information Processing Systems, vol. 5*, San Mateo, CA, 1993. Morgan Kaufmann.

[19] Neil J. Bershad and Lian Zuo Qu. On the probability density function of the lms adaptive filter weights. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:43–56, 1989.

[20] V. Fabian. On asymptotic normality in stochastic approximation. *Ann. Math. Statist.*, 39:1327–1332, 1968.

[21] Halbert White. Learning in artificial neural networks: A statistical perspective. *Neural Computation*, 1:425–464, 1989.

[22] Larry Goldstein. Mean square optimality in the continuous time Robbins Monro procedure. Technical Report DRB-306, Dept. of Mathematics, University of Southern California, LA, 1987.