EUROPHYSICS LETTERS

Europhys. Lett., (), pp. ()

On–line learning from a finite training set: A solvable model

B. López[1] and M. Opper[2]

[1] *Institut für Theoretische Physik, Julius–Maximilians–Universität Würzburg, D–97074 Würzburg, Germany*

[2] *Neural Computing Research Group, Aston University, B4 7ET Birmingham, UK*

(received ; accepted)

PACS. 87.10+e – General, theoretical, and mathematical biophysics, including logic of biosystems, quantum biology, and relevant aspects of thermodynamics, information theory, cybernetics, and bionics.
PACS. 02.50−r – Probability theory, stochastic processes, and statistics.
PACS. 07.05.Mh– Neural networks, fuzzy logic, artificial intelligence.
PACS. 05.20.-y – Statistical mechanics.

Abstract. – We discuss the problem of on–line learning from a finite training set with feedforward neural networks. Defining a modified learning rule, which randomly chooses inputs and weights to be updated, the dynamics of learning can be treated within a diffusion approximation in the thermodynamic limit. No assumption on the generation of data is made. Explicit results for the stationary distribution and relaxation times can be found for a network with linear transfer function. Assuming selfaveraging of the diffusion term, a general relation between on–line learning and batch learning with an effective temperature is established.

Feedforward architectures [1] are among the most popular types of neural networks currently under study. Using methods of statistical mechanics, many attempts have been made to understand their ability to learn and generalize from examples. Mostly, in these approaches it is assumed that the learning algorithm is based on the minimization of a training energy which is an additive function of *all* examples in a training set. The effect of additive noise (temperature) is easily incorporated. However, many algorithms which are used in practise will not directly fit into this framework. Training is often done in an online way [2, 3, 4] where only *one example* at each time step is used for an update. Usually, we can distinguish between two different online learning scenarios: For the first one, the network must adapt in every time step to a completely new random example. By its Markovian nature, this stochastic learning process allows a rather thorough analysis [2, 3]. In the thermodynamic limit of large networks, under the assumptions of nice and symmetric probability distributions of examples, the introduction of order parameters allows an exact description of the learning dynamics even for complicated multilayer networks [**?**, 9].

In the second online scenario, the presented example belongs to a training set of finite size and may thus appear more than once during the training phase [4, 5, 6, 7]. Since the dynamics

Typeset using EURO-TEX

is no longer Markovian with respect to the generation of the examples, general results for this practically important sceanrio are hard to obtain and are usually restricted to the case of small learning rates [4]. Recent attempts to study this problem in the order parameter framework of the thermodynamic limit have sofar succeeded for a network with linear transfer function [8] and the Perceptron with a Hebbian learning rule.[13] Approximate approaches for nonlinear networks within this framework can be found in [11].

The goal of this letter is to present a modified stochastic update rule for online learning with a fixed training set which enables us to establish general relations between the online scenario and the batch scenario with a temperature. This will make the online scenario accessible to methods of equilibrium statistical mechanics. In contrast to the orderparameter approach, we treat the training data as fixed, nonrandom quantities and average only over the randomness of the update. Hence, the thermodynamic limit will be used only in a rather weak form. At a later stage, for specific cases, we may use equilibrium methods (replica trick, high temperature expansions, etc.) in order to perform further averages over the examples.

For simplicity, we will consider a single–layer network with N–dimensional weight vector $\vec{\omega}$. Generalizations to two layer networks with a fixed hidden to ouput function (committee, parity machines) is possible and will be given elsewhere. The training error, or energy, for a given set of examples $\{\vec{\xi}^{\mu}, \sigma^{\mu}\}_{\mu=1}^{p}$ is

$$E(\vec{\omega}) = \sum_{\mu=1}^{p} \left(\sigma_{\mu} - g(h_{\mu})\right)^2 \ , \tag{1}$$

where σ_{μ} is the desired output for pattern $\vec{\xi}^{\mu}$ and $g(h)$ is the transfer function depending on the internal field $h^{\mu} = \vec{\omega} \cdot \vec{\xi}^{\mu}/\sqrt{N}$. For simplicity, we concentrate on binary inputs $\xi_i^{\mu} = \pm 1$, so that $\vec{\xi}^{\mu} \cdot \vec{\xi}^{\mu} = N$. The basic prototype of most learning lagorithms, ie. Backpropagation [12], consists in its batch version of an update of all weights $\omega_i(\tau + 1) = \omega_i(\tau) + \Delta\omega_i(\tau)$ at time step τ by gradient descent of the training energy $E(\vec{\omega})$. Here, in every time step all p patterns are used for the update. Mathematically, we can switch to an online scenario by introducing a decision variable $\gamma^{\mu}(\tau) = \{0, 1\}$ which is 1 for the example to be used in the update and 0 for the others. E.g. $\gamma^{\mu}(\tau) = \delta_{\tau \bmod p,(\mu-1)}$, for a cyclic update through the training set. A randomized choice consists in choosing $\gamma^{\mu}(\tau)$ to be 1 (example is used for update) or 0 (not used) at random with probability γ and $(1 - \gamma)$ respectively. In such a case, γp examples are used on average for each step. In our approach, we go one step further in randomization by introducing the decision variable $\gamma_i^{\mu}(\tau) = \{0, 1\}$, which also selects the sites i at random for which the weights are updated at time step τ with an example μ. The update is defined as

$$\Delta\omega_i(\tau) = \eta \sum_{\mu} \epsilon_{\mu} \frac{1}{\sqrt{N}} \xi_i^{\mu} \gamma_i^{\mu}(\tau) \ . \tag{2}$$

where $\epsilon_{\mu} = (\sigma_{\mu} - g(h_{\mu}))g'(h_{\mu})$. The $\gamma_i^{\mu}(\tau)$ are independent random variables for all i, μ and τ which take the values 1 or 0 with probability γ and $(1 - \gamma)$, respectively. The batch case is recovered for $\gamma = 1$, whereas for $\gamma = 1/p$ a weight i is updated on average with one pattern per time step. Before discussing the general case, we first give a result for the asymptotic energy averaged over the distribution of the random variables $\gamma_i^{\mu}(\tau)$ for the linear transfer function $g(h) = h$. Using $g'(h) = 1$, linear equations of motion for $\epsilon_{\mu}(\tau) = \sigma_{\mu} - h_{\mu}(\tau)$, can be derived, which can be solved in closed form. Details will be given elsewhere. As a result, the asymptotic on–line energy can be expressed by

$$\overline{E(\tau \to \infty)} = \frac{1}{1 - \delta} \ E_0 \ , \tag{3}$$

where E_0 is the asymptotic energy of the batch algorithm (the minimum of (1)) and

$$\delta = \eta(1-\gamma)N^{-1}\sum_{\lambda}\left(2-\eta\gamma\lambda\right)^{-1} . \tag{4}$$

The sum runs over all except the zero eigenvalues of the pattern correlation matrix $C = N^{-1}\sum_{i=1}^{N}\xi_i^\mu\xi_i^\nu$. Necessary conditions for the convergence are $\eta\gamma\lambda_{\max} < 2$ and $\delta < 1$, which implicitly defines a maximal learning rate. Note, that this result holds for any value of the on–line parameter γ, for all N and a fixed training set, which is characterized by the eigenvalues of the correlation matrix. For $\alpha = p/N \leq 1$ the batch energy will be zero and our stochastic on–line learning rule converges also to zero training energy. Since $\delta > 0$, we find that for a non-learnable scenario ($E_0 > 0$), (2) does not reach the minimum of the energy. The couplings will keep on fluctuating around the minimum forever. Only if the learning rate η tends to zero, the batch minimum will be reached. We may understand this effect by observing, that in the nonlearnable case not all p linear errors ϵ_μ will become zero asymptotically. If some of the $\gamma_i^\mu(\tau)$ are zero, (2) does not compute the correct gradient of the energy and the update $\Delta\omega_i(\tau)$ will always stay nonzero.

If we take the thermodynamic limit $N \to \infty$ such that $\alpha = p/N$ together with the on–line limit $\gamma = 1/p$, (4) simplifies to $\lim_{\gamma\to 0}\ \delta = \frac{\eta}{2}$, independently of the training set! The resulting upper limit of the learning rate $\eta = 2$ is also known for related iterative algorithms for linear systems of equations [17] and was also obtained in online learning from an infinite training set [19].

The general analysis of the online algorithm (2) is based on the fact that the update rule (2) defines a Markov–process with respect to the random choice of the $\gamma_i^\mu(\tau)$ for which the entire statistics is determined by its transition probability $T(\vec{\omega}|\vec{\omega}')$. For $p, N \to \infty$ we can apply the central limit theorem to (2) and show that for all γ, $T(\vec{\omega}|\vec{\omega}')$ becomes a multivariate Gaussian distribution with mean $\gamma\vec{a}$ and diagonal (!) covariance matrix having equal elements $(\gamma-\gamma^2)D$, where

$$\vec{a} = -\frac{\eta}{2}\nabla E(\omega) = \eta\sum_{\mu}\epsilon_\mu\frac{1}{\sqrt{N}}\vec{\xi^\mu} \qquad \text{and} \qquad D = \eta^2\frac{1}{N}\sum_{\mu}\epsilon_\mu^2 . \tag{5}$$

The analyis simplifies further in the on–line limit $\gamma = 1/p \to 0$, where both mean and variances scale $\propto \gamma$. Introducing a continuous time scale via $t = \gamma\tau$, we can replace the dynamics by a diffusion process for which the probability density of the couplings satisfies the Fokker–Planck–equation (FPE) [15]

$$\frac{\partial P(\vec{\omega},t)}{\partial t} = \left[-\sum_i\frac{\partial}{\partial\omega_i}a_i(\vec{\omega}) + \frac{1}{2}\sum_i\frac{\partial^2 D(\vec{\omega})}{\partial\omega_i^2}\right]P(\vec{\omega},t) . \tag{6}$$

Note, that our diffusion limit is entirely based on the thermodynamic limit rather than on the assumption of small learning rates as in [4, 5, 6, 7]. For the linear case $g(h) = h$, the stationary distribution with $\partial P_{\rm stat}(\vec{\omega})/\partial t = 0$ can be calculated explicitely (the diffusion term D equals $\eta^2 E(\vec{\omega})/N$ and the drift $\vec{a}$ is proportional to the gradient of the diffusion such that a certain potential condition [15] is fulfilled), yielding

$$P_{\rm stat}(\vec{\omega}) = C_N\ E(\vec{\omega})^{-\frac{N}{\eta}} , \tag{7}$$

with C_N a normalization constant. $P_{\rm stat}$ is not a Boltzmann–Gibbs distribution, as would be the case for a Gaussian additive noise term in (2). Similar distributions were found previously

in related special cases [6, 7, 18]. From (7) we find the energy distribution

$$P(E) \propto \left(\frac{E_0}{E^2}\right)^{\frac{(2-\eta)N}{2\eta}} \left(1-\frac{E_0}{E^2}\right)^{\frac{N}{2}} \Theta(E-E_0) \ , \tag{8}$$

with E_0 the asymptotic value of the batch energy. Although the thermodynamic limit of large N and p has been used to derive (6), this equation still contains nontrivial dependence on N. Figure 1 displays $P(E)$ for a training set with $N = 20$ compared to results from simulations, showing good agreement already for this value of N. The width of the distribution $P(E)$ decreases as $N^{-1/2}$, so that for $N \to \infty$ it becomes a delta function at $\overline{E}$.

From the equilibrium distribution also the asymptotic average generalization error can be inferred

$$\epsilon_g = \left\langle \overline{\left(\sigma_\mu - h_\mu(\vec{\xi})\right)^2} \right\rangle = \epsilon_g^{batch} + \frac{\eta}{2-\eta} E_0 \frac{1}{N} \mathrm{Tr}\left(B^{-1}\widetilde{B}\right) > \epsilon_g^{batch} \ . \tag{9}$$

The additional average $\langle\cdot\rangle$ is over the distribution of test patterns $\vec{\xi}$ where $\widetilde{B}_{ij} = \langle \xi_i^\mu \xi_j^\mu \rangle$ and $B_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu$, is the empirical spatial correlation matrix of the training patterns. This is interesting, because in nonlearnable scenarios a certain amount of additive noise in the training process, which prevents the training energy from relaxing into the minimum has proved to be helpful in avoiding overfitting [20]. In our case however, the state dependent multiplicative noise generated by the online process itself leads always to a decrease in performance!

In order to extend our results to general nonlinear output units $g(h)$, we may realize, that we have not fully utilized the properties of the thermodynamic limit. The state dependent diffusion $D = \eta^2 \frac{1}{N}\sum_\mu \epsilon_\mu^2$ is an intensive quantity which becomes selfaveraging in this limit and a mean field aproximation which replaces D by its average $\overline{D}$, should become exact. For constant diffusion term $\overline{D}$, the resulting stationary distribution is of the standard Boltzmann–Gibbs type $P_{\rm stat}(\vec{\omega}) \propto \exp\left(-\beta E(\vec{\omega})\right)$ with a *temperature* $T = 1/\beta$ which has to be determined selfconsistently via

$$\eta/\beta = \overline{D} = \frac{\eta^2}{N} \int \sum_\mu \epsilon_\mu^2(\vec{\omega}) P_{\rm stat}(\vec{\omega}) d\vec{\omega}. \tag{10}$$

Hence, in the thermodynamic limit, we have mapped the statistics of our online algorithm with a fixed training set onto a problem of batch learning with additive noise. In order to illustrate this result, we consider a single layer perceptron with transfer function $g(h) = \mathrm{sign}(h)$. Learning algorithms for this noncontinuous function are not based on the training energy (1) but one can use functions of the form

$$E(\vec{\omega}) = \sum_\mu \left(\kappa - \sigma_\mu h_\mu\right)^m \Theta(\kappa - \sigma_\mu h_\mu) \ . \tag{11}$$

instead. $m = 1$ corresponds to Rosenblatt's perceptron learning algorithm and $m = 2$ to the Adatron rule [10]. The online version of the algorithms is also of the form (2). For $m = 1$ one has $\epsilon_\mu = \frac{1}{2}(\kappa - \sigma_\mu h_\mu)\Theta(\kappa - \sigma_\mu h_\mu)$. Figure 2 shows the average inverse temperature from simulations of the algorithm with $m = 1$ using random outputs and a spherical distribution of inputs together with the theoretical prediction based on the mean field assumption (10). The latter was obtained by a replica calculation of the equilibrium properties with Hamiltonian (11) and a selfconsistent solution for the temperature (10). A more general result can be obtained for the case $m = 2$ which leads to almost the same FPE as for the linear output unit. Again, in the nonlearnable case the static distribution is of the form (7)!

A general treatment of the relaxation properties of the algorithm (2) into the stationary state is yet unclear. However, we may get some insight into this problem by returning to the linear case. Instead of investigating the FPE, it is more convenient to look at the corresponding stochastic differential equation using Ito–calculus [16]. We find that the time evolution of the non–diagonal elements of $\overline{\omega_i(t)\omega_j(t)}$ is not altered with respect to the batch dynamics, whereas the decay of the diagonal elements to the the fixpoint $\vec{y}^*$ for large times $t > T \gg 1$ is given by

$$\overline{\omega_i^2(t)} \simeq \vec{y}^* + e^{-2\eta A t}\vec{v}_1(T) + e^{-\eta\lambda_1 t}\vec{v}_2 \ . \tag{12}$$

$\vec{v}_2$ is a constant vector and $\vec{v}_1$ depends only on the fixed time T. The matrix A is given by $A_{ij} = \lambda_i\delta_{ij} - \frac{\eta}{2N}\lambda_j$, with λ_i the eigenvalues of the input correlation matrix B, λ_1 being the smallest. For $\eta = 0$ (batch case) $A = B$. For $\eta > 0$ the asymptotic relaxation time will be different from the batch case, only if the smallest eigenvalue α_1 of A becomes less than $\lambda_1/2$. In figure 3, a typical evolution of the eigenvalues of A as a function of η is depicted. For η close to 2, α_1 approaches zero as $\alpha_1(\eta \to 2) \simeq (1-\eta/2)/(N^{-1}\sum_i \lambda_i^{-1})$ and the learning time diverges according to $\tau(\eta \to 2) \sim (2-\eta)^{-1}$.

Inserting $\alpha_1 = \lambda_1/2$ into the characteristic polynomial of the matrix A and setting the result to zero, we obtain the critical learning rate η_c, above which the relaxation is slower than for batch learning:

$$\eta_c = 2N \ \frac{\prod_{i=1}^N (\lambda_i - \lambda_1/2)}{\sum_{i=1}^N \lambda_i \prod_{k\neq i} (\lambda_k - \lambda_1/2)} \ . \tag{13}$$

Hence, the optimal learning rate is $\eta_{opt} = \eta_c$ and the corresponding learning time $\tau_{opt} = (\eta_c\lambda_1)^{-1}$.

To summarize, we have introduced a modified dynamics for online learning from a finite training set, which allows us to derive exact relations between online learning and batch learning with a temperature. It is not hard to extend our equilibrium approach with effective temperature (10) to one of the standard models of two layer networks with fixed hidden to output weights. The main reason, why the stationary state of the online dynamics can be mapped onto an effective Boltzmann–Gibbs equilibrium is the fact that the Fokker–Planck equation (6) has only diagonal elements in the diffusion matrix. Nondiagonal elements would appear in the more common online scenario where the random decision variables $\gamma^\mu(\tau)$ are the same for all weights. It will be interesting to try a perturbative treatment of these nondiagonal elements in order to see to what extent their influence can also be incorporated within an effective temperature. Recent simulations on multilayer networks [21] seem to indicate that a description in terms of an effective thermal equilibrium may in fact be reasonable in a variety of cases.

We would like to thank M. Biehl, R. Urbanczik and G. Reents for useful discussions. This work has been supported by the grant Op 45/5-2 of the Deutsche Forschungsgemeinschaft.

REFERENCES

[1] J. A. Hertz, A. Krogh and R. G. Palmer 1991 *Introduction to the Theory of Neural Computation* (Redwood City, CA: Addison–Wesley)
[2] S. Amari 1993 *Neurocomputing* **5** 185–196
[3] M. Biehl and H. Schwarze 1995 *J. Phys. A* **28** 643–656
[4] T. Heskes and B. Kappen 1991 *Phys. Rev.* A **44** 2718–2762
[5] T. Heskes 1994 *J. Phys. A* **27** 5145–5160

[6] G. Radons 1993 *J. Phys. A* **26** 3455–3461
[7] L. Hansen, R. Pathria and P. Salamon 1993 *J. Phys. A* **26** 63–71
[8] P. Sollich and D. Barber 1997 *Europhys. Lett.* **38** 477-482
[9] D. Saad and S. Solla 1995 *Phys. Rev. Lett.* **74** 4337 and *Phys. Rev.* E **52** 4225
[10] J. K. Anlauf and M. Biehl 1990 *Europhys. Lett.* **10** 687
[11] D. Barber and P. Sollich 1998 in [14]
[12] D. E. Rumelhart, G. E. Hinton and R. J. Williams 1986 *Nature* **323** 533–536
[13] A. C. C. Coolen and D. Saad 1998 in [14]
[14] D. Saad (editor) 1998 *On–Line Learning in Neural Networks* (Cambridge: University Press)
[15] J. Honerkamp 1994 *Stochastic Dynamical Systems* (New York: VCH Publishers)
[16] C. W. Gardiner 1990 *Handbook of Stochastic Methods* (Berlin: Springer–Verlag)
[17] J. Stoer and R. Bulirsch 1990 *Numerische Mathematik 2* (Berlin: Springer–Verlag)
[18] T. Leen and J. Moody 1992 *Advances in Neural Information Processing Systems 5* ed S. Hanson *et al* (San Mateo: Morgan Kaufmann) 451-458
[19] M. Biehl, private communication
[20] S. Bös and M. Opper 1997 *Advances in Neural Information Processing Systems 9* ed C.M. Mozer *et al* (Cambridge: MIT Press)
[21] M. Ahr, M. Biehl and R. Urbanczik 1999 *Eur. Phys. J.* B to appear