*Europhys. Lett.*, **51** (6), pp. 691–697 (2000)

# On-line learning from restricted training sets in multilayer neural networks

A. C. C. COOLEN[1], D. SAAD[2] and YUAN-SHENG XIONG[2]

[1] *Department of Mathematics, King's College - Strand, London WC2R 2LS, UK*
[2] *The Neural Computing Research Group, Aston University - Birmingham B4 7ET, UK*

(received 7 February 2000; accepted in final form 25 July 2000)

PACS. 87.10.+e – Biological and medical physics: General theory and mathematical aspects.
PACS. 02.50.-r – Probability theory, stochastic processes, and statistics.
PACS. 05.90.+m – Other topics in statistical physics, thermodynamics, and nonlinear dynamical systems.

**Abstract.** – We analyse the dynamics of on-line learning in multilayer neural networks where training examples are sampled with repetition and where the number of examples scales with the number of network weights. The analysis is based on monitoring a set of macroscopic variables from which the training and generalisation errors can be calculated. A closed set of dynamical equations is derived using the dynamical replica method and is solved numerically. The theoretical results are consistent with those obtained by computer simulations.

Layered neural networks are powerful nonlinear information processing systems, capable of implementing arbitrary continuous and discrete input-output maps to any desired accuracy [1], given a sufficient number of hidden nodes and a sufficiently large example set. They have been employed successfully in a variety of regression and classification tasks, and have been studied using a wide range of methods (for a review, see [2]). On-line learning refers to the iterative modification of the network parameters according to a predetermined training rule, following successive presentations of single training examples, each representing a specific input vector and the corresponding output. On-line learning is one of the leading techniques in training large neural networks, especially via gradient descent on a differentiable error measure.

Considerable progress has been made recently in analysing the dynamics of supervised on-line learning in layered neural networks via methods of statistical physics (reviews can be found in [3] and [4]). Most of the analyses (*e.g.*, [5–7]) have concentrated on the case of uncorrelated infinite training sets, where training examples are sampled without repetition and in which there is no correlation between the network parameters and the examples presented at each training step. However, a more realistic scenario is that where the number of training examples scales with the number of free parameters, and where examples are sampled with repetition. This gives rise to correlations between the network parameters and the training examples, which clearly affect the learning process. One of the most significant aspects of having a fixed example set is the distinction between the two key performance measures: the *training error*, measuring the network performance with respect to the restricted training set, and the *test (generalisation) error*, calculated for all possible inputs sampled from the true distribution. The former may be monitored in practical training scenarios, while the latter (the reduction of which is the true aim of the learning process) can only be assessed up to some confidence level.

The study of learning from fixed example sets [8–12] has been mostly restricted to single-layer systems, focusing on specific (usually simple) learning rules or on binary weights. In addition, most of these studies were restricted to batch learning, where the network parameters are modified only after the full example set has been presented. We recently proposed a new approach to the analysis of on-line learning from restricted training sets, based on the dynamical replica method, which enables one to deal with arbitrary training rules and which can treat both on-line and batch learning scenarios [13]. In this letter we extend the analysis to the case of on-line learning in multilayer networks and obtain numerical solutions in specific scenarios. We then validate the theoretical results against numerical simulations. For brevity we will not consider here the case of batch learning.

We concentrate on information processing tasks in the form of maps from an $N$-dimensional input space $\boldsymbol{\xi} \in I\!R^N$ onto a scalar $\zeta \in I\!R$, realized through a parametrised function $\sigma(\boldsymbol{J}, \boldsymbol{\xi}) = \sum_{i=1}^{K} g\,(\boldsymbol{J}_i \cdot \boldsymbol{\xi})$. This function can be viewed as a two-layer neural network, where $g$ is the activation function of the hidden units, taken here to be the error function $g(x) \equiv \mathrm{erf}(x/\sqrt{2})$, $\boldsymbol{J} \equiv \{\boldsymbol{J}_i\}_{1 \le i \le K}$ is the set of input-to-hidden adaptive weights for the $K$ hidden nodes, and the hidden-to-output weights are set to 1. The activation of hidden node $i$ under presentation of the input pattern $\boldsymbol{\xi}^\mu$ is denoted $x_i^\mu = \boldsymbol{J}_i \cdot \boldsymbol{\xi}^\mu$. This general configuration, usually referred to as the "soft committee machine" [6], encompasses most of the properties of general multilayer networks. Training examples are drawn from a finite set $\tilde{D} = \{\boldsymbol{\xi}^1, .., \boldsymbol{\xi}^p\}$ and are of the form $(\boldsymbol{\xi}^\mu, \zeta^\mu)$, where $\mu = 1, 2, .., p$. The components of the independently drawn input vectors $\boldsymbol{\xi}^\mu$ are uncorrelated random variables with zero mean and unit variance. The corresponding output $\zeta^\mu$ is given by a deterministic teacher of an architecture similar to that of the student, except for a possible difference in the number $M$ of hidden units: $\zeta^\mu = \sum_{n=1}^{M} g\,(\boldsymbol{B}_n \cdot \boldsymbol{\xi}^\mu)$, where $\boldsymbol{B} \equiv \{\boldsymbol{B}_n\}_{1 \le n \le M}$ is the set of input-to-hidden adaptive weights for teacher hidden nodes. The activation of hidden node $n$ under presentation of the input pattern $\boldsymbol{\xi}^\mu$ is denoted $y_n^\mu = \boldsymbol{B}_n \cdot \boldsymbol{\xi}^\mu$. The indices $i, j, k, \ldots$ refer to units in the student network and $n, m, \ldots$ to units in the teacher network. Sums over the various indices will be considered from 1 to $K$ or to $M$, respectively. The general framework [13] allows for the analysis of any training rule $\boldsymbol{\mathcal{G}}$ such that the network parameters are modified in the following manner:

$$\boldsymbol{J}_j^{l+1} \;=\; \boldsymbol{J}_j^l + \frac{\eta}{N} \boldsymbol{\xi}(l)\, \mathcal{G}_j[\boldsymbol{x}(l), \boldsymbol{y}(l)]\,, \tag{1}$$

where $l$ represents the current time step in which a single example is randomly drawn from $\tilde{D}$ and invokes the parameter update.

Here we concentrate on the most common on-line learning scenario for regression tasks, where the function $\boldsymbol{\mathcal{G}}$ is the gradient with respect to the parameters $\boldsymbol{J}$ of the quadratic error measure (per example): $E(\boldsymbol{J}, \boldsymbol{\xi}) = [\,\sigma(\boldsymbol{J}, \boldsymbol{\xi}) - \zeta\,]^2\,/2$.

The fundamental difference between the infinite and restricted training set scenarios is that the joint probability distribution for the student and teacher node activations $\boldsymbol{x}$ and $\boldsymbol{y}$, which is Gaussian in the infinite training set case, takes here a more general form, which depends on the training patterns and changes dynamically during the learning process. In fact, it seems to be quite natural to define this joint probability as one of the macroscopic variables to be monitored continuously [13],

$$P(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{J}) = \frac{1}{p} \sum_{\mu} \prod_{i} \delta(x_i - \boldsymbol{J}_i \cdot \boldsymbol{\xi}^\mu) \prod_{n} \delta(y_n - \boldsymbol{B}_n \cdot \boldsymbol{\xi}^\mu)\;, \tag{2}$$

together with the overlaps $R_{in}(\boldsymbol{J}) = \boldsymbol{J}_i \cdot \boldsymbol{B}_n$ (between student and teacher weight vectors) and $Q_{ij}(\boldsymbol{J}) = \boldsymbol{J}_i \cdot \boldsymbol{J}_j$ (between student weight vectors). We also define the constant overlap matrix

$T_{nm} = \boldsymbol{B}_n \cdot \boldsymbol{B}_m$. In the thermodynamic limit $N \to \infty$ the macroscopic variables $\{Q, R, P\}$ are found to evolve deterministically in time, and are sufficient for calculating the two main performance measures: the generalisation error, which corresponds to averaging $E(\boldsymbol{J}, \boldsymbol{\xi})$ over the Gaussian input distribution, providing the same expression as in the infinite training set case [7]; and the training error, using the abbreviation $\langle f(\boldsymbol{x}, \boldsymbol{y}) \rangle = \int \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}\ P(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{x}, \boldsymbol{y})$,

$$E_t = \left\langle \frac{1}{2} \left[ \sum_{i=1}^{K} g(x_i) - \sum_{n=1}^{M} g(y_n) \right]^2 \right\rangle . \tag{3}$$

To follow the dynamics, one derives a set of coupled differential equations [13] describing the evolution of the macroscopic variables in the limit $N \to \infty$:

$$\frac{\mathrm{d}}{\mathrm{d}t} Q = \eta (V + V^T) + \eta^2 Z , \qquad \frac{\mathrm{d}}{\mathrm{d}t} R = \eta W \tag{4}$$

and

$$\begin{aligned} \frac{\partial}{\partial t} P(\boldsymbol{x}, \boldsymbol{y}) \ = \ & \frac{1}{\alpha} \int \mathrm{d}\boldsymbol{x}'\, P(\boldsymbol{x}', \boldsymbol{y}) \left[ \prod_i \delta[x_i - x_i' - \eta\, \mathcal{G}_i(\boldsymbol{x}', \boldsymbol{y})] - \prod_i \delta(x_i - x_i') \right] - \\ & - \eta \sum_i \frac{\partial}{\partial x_i} \int \mathrm{d}\boldsymbol{x}' \mathrm{d}\boldsymbol{y}' \mathcal{G}_i(\boldsymbol{x}, \boldsymbol{y}) \mathcal{A}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{x}', \boldsymbol{y}') + \frac{\eta^2}{2} \sum_{i,k} Z_{ik} \frac{\partial^2 P(\boldsymbol{x}, \boldsymbol{y})}{\partial x_i \partial x_k} , \end{aligned} \tag{5}$$

defining the matrices $V = \langle \boldsymbol{\mathcal{G}} \boldsymbol{x}^T \rangle$, $W = \langle \boldsymbol{\mathcal{G}} \boldsymbol{y}^T \rangle$, and $Z = \left\langle \boldsymbol{\mathcal{G}} \boldsymbol{\mathcal{G}}^T \right\rangle$. This set of equations cannot be closed in general; the difficulties originate in the Green's function

$$\mathcal{A}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{x}', \boldsymbol{y}') = \langle \delta(\boldsymbol{x} - \boldsymbol{J} \cdot \boldsymbol{\xi}) \delta(\boldsymbol{y} - \boldsymbol{B} \cdot \boldsymbol{\xi}) (1 - \delta_{\boldsymbol{\xi}\boldsymbol{\xi}'}) \boldsymbol{\xi} \cdot \boldsymbol{\xi}' \delta(\boldsymbol{x}' - \boldsymbol{J} \cdot \boldsymbol{\xi}') \delta(\boldsymbol{y}' - \boldsymbol{B} \cdot \boldsymbol{\xi}') \rangle_{p_t(\boldsymbol{J}|QRP)} ,$$

where the average is with respect to $p_t(\boldsymbol{J}|QRP)$, the weight probability density conditioned on the values of the macroscopic observables $\{Q, R, P\}$ at time $t$ (the microscopic measure in macroscopic sub-shells of the ensemble). We follow the derivation of [13] and employ the dynamical replica theory [14] to close eqs. (4), (5) by making two key assumptions:

i) For $N \to \infty$ the macroscopic observables obey *closed* dynamic equations; more specifically, we may thus assume equipartitioning of probability (or maximum entropy) in the macroscopic sub-shells:

$$p_t(\boldsymbol{J}|QRP) \sim \prod_{i,k} \delta[Q_{ik} - Q_{ik}(\boldsymbol{J})] \prod_{i,n} \delta[R_{in} - R_{in}(\boldsymbol{J})] \prod_{\boldsymbol{x},\boldsymbol{y}} \delta[P(\boldsymbol{x}, \boldsymbol{y}) - P(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{J})] . \tag{6}$$

ii) The macroscopic equations are self-averaging with respect to the specific realisation of $\tilde{D}$; this allows for the averaging of the macroscopic variables over all training sets.

Both assumptions can be regarded as good approximations in general and will be validated against simulation results. They may become exact in some cases (*e.g.*, Hebbian learning); we believe the second assumption to be exact in general. Following the calculation of [13] and employing the replica identity,

$$\left\langle \frac{\int \mathrm{d}\boldsymbol{J} W[\boldsymbol{J}, z] G[\boldsymbol{J}, z]}{\int \mathrm{d}\boldsymbol{J} W[\boldsymbol{J}, z]} \right\rangle_z = \lim_{n \to 0} \int \mathrm{d}\boldsymbol{J}^1 \cdots \mathrm{d}\boldsymbol{J}^n \left\langle G[\boldsymbol{J}^1, z] \prod_{\alpha=1}^{n} W[\boldsymbol{J}^\alpha, z] \right\rangle_z , \tag{7}$$

one obtains, under the further assumption of replica symmetry, a closed form for eq. (5):

$$\frac{\partial}{\partial t}P(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{\alpha}\int \mathrm{d}\boldsymbol{x}'\ P(\boldsymbol{x}',\boldsymbol{y})\left[\prod_i \delta[x_i - x_i' - \eta\mathcal{G}_i(\boldsymbol{x}',\boldsymbol{y})] - \prod_i \delta(x_i - x_i')\right] -$$
$$-\eta\sum_i \frac{\partial}{\partial x_i}\left[[W\boldsymbol{y} + U(\boldsymbol{x} - R\boldsymbol{y}) + X(Q - RR^T)\Phi(\boldsymbol{x},\boldsymbol{y})]_i P(\boldsymbol{x},\boldsymbol{y})\right] + \frac{\eta^2}{2}\sum_{ik} Z_{ik}\frac{\partial^2 P(\boldsymbol{x},\boldsymbol{y})}{\partial x_i \partial x_k}, \quad (8)$$

where we have introduced the matrices $B = (Q-q)^{-1}L$, $X = (V - WR^T)(Q - RR^T)^{-1} - U$, $LL^T = q - RR^T$, $U = \langle \mathcal{G}\Phi^T \rangle$, and where

$$\Phi_i(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{P(\boldsymbol{x}|\boldsymbol{y})}\int D\boldsymbol{z}\left\langle [(Q-q)^{-1}(\boldsymbol{x} - \boldsymbol{x}')]_i \right\rangle_* \langle \delta(\boldsymbol{x} - \boldsymbol{x}')\rangle_*, \quad (9)$$

using the notation $D\boldsymbol{z} \equiv \prod_{i=1}^K 1/\sqrt{2\pi}\ e^{-\frac{1}{2}z_i^2}\ \mathrm{d}z_i$ and

$$\langle f(\boldsymbol{x},\boldsymbol{x}')\rangle_* = \frac{\int \mathrm{d}\boldsymbol{x}'\ M(\boldsymbol{x}',\boldsymbol{y})\ e^{\boldsymbol{x}'^T B\boldsymbol{z}} f(\boldsymbol{x},\boldsymbol{x}')}{\int \mathrm{d}\boldsymbol{x}'\ M(\boldsymbol{x}',\boldsymbol{y})\ e^{\boldsymbol{x}'^T B\boldsymbol{z}}}\ . \quad (10)$$

The $K \times K$ matrix $q$ and the function $M(\boldsymbol{x}',\boldsymbol{y})$ emerge from the replica calculation and relate to the cross-replica overlap matrix $Q$ and the conjugate variable to $P(\boldsymbol{x}|\boldsymbol{y})$, respectively. They are calculated at each step by solving the saddle point equations (for details, see [13]).

Although eq. (8) can be calculated at each time step, leading to a solution of eqs. (4) and (8), this will clearly come at a high computational cost, which is daunting already in the case of single-layer networks. Obtaining such solutions in the case of multilayer networks is clearly infeasible. We therefore resort to the large-$\alpha$ approximation which was shown to provide a highly accurate approximated solutions in the single-layer case even for low $\alpha$ values (as low as $\alpha = 0.5$). This enables one to obtain the simple following form for eq. (8) without solving a set of saddle point equations at each time step (using the notation $\bar{\boldsymbol{x}}(\boldsymbol{y}) = \int \mathrm{d}\boldsymbol{x}\ \boldsymbol{x}\ P(\boldsymbol{x}|\boldsymbol{y})$):

$$\frac{\partial}{\partial t}P(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{\alpha}\int \mathrm{d}\boldsymbol{x}'\ P(\boldsymbol{x}',\boldsymbol{y})\left[\prod_i \delta[x_i - x_i' - \eta\mathcal{G}_i(\boldsymbol{x}',\boldsymbol{y})] - \prod_i \delta(x_i - x_i')\right] -$$
$$-\eta\sum_i \frac{\partial}{\partial x_i}\left[\Gamma_i(\boldsymbol{x},\boldsymbol{y})P(\boldsymbol{x},\boldsymbol{y})\right] + \frac{\eta^2}{2}\sum_{i,k} Z_{ik}\frac{\partial^2 P(\boldsymbol{x},\boldsymbol{y})}{\partial x_i \partial x_k}\ , \quad (11)$$

where

$$\Gamma_i(\boldsymbol{x},\boldsymbol{y}) = \left[\begin{pmatrix} V \\ W \end{pmatrix}^T \begin{pmatrix} Q & R \\ R^T & T \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} - [\langle \mathcal{G}\bar{\boldsymbol{x}}^T(\boldsymbol{y})\rangle - WR^T](Q - RR^T)^{-1}[\bar{\boldsymbol{x}}(\boldsymbol{y}) - R\boldsymbol{y}]\right]_i .$$

This approximation is particularly suitable to the model examined here, since the main features of learning in multilayer networks, such as the breaking of internal symmetries and the asymptotic convergence, can be observed at sensible time scales only for high $\alpha$ values.

The dynamical equations (4), (11) are the main results of the large-$\alpha$ approximation, and can be solved in principle to provide rather accurate approximated solutions. However, obtaining the solutions is difficult in the case of multi-layer neural networks as one should monitor the evolution of a multivariate probability distribution, which does not have a clear analytical form; and solve numerically the differential equations (11) and (4).

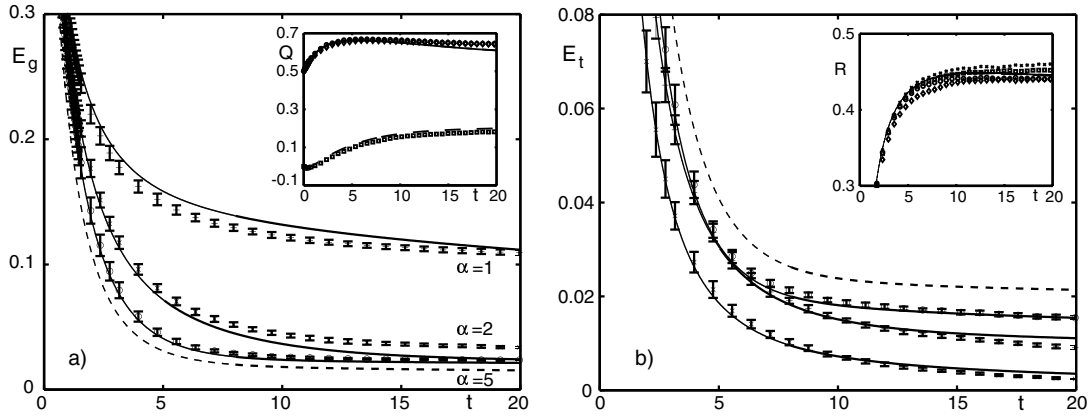


Fig. 1 – The evolution of the generalisation (a) and training errors (b) as a function of time for $\alpha = 1, 2, 5$ ($\eta = 0.5$). The solid lines represent analytical results while simulation experiments are presented by symbols; both were initialised in a similar manner. Simulation results were averaged over 20 trials. Theoretical results for the training and generalisation errors in the case of $\alpha = 5$ are presented in (a) and (b), respectively for comparison (dashed line). The insets in both figures show the evolution of the various overlaps ($Q$ and $R$, respectively, different symbols represent the various overlaps) in the case of $\alpha = 5$, comparing theoretical results and simulations (mean values).

To make the calculation feasible in the case of multilayer networks, we look for a parametric approximated representation of the probability distribution. We have considered two different possibilities: a mixture of multivariate Gaussian distributions and the local Gaussian approximation, whereby the conditional probability $P(\boldsymbol{x}|\boldsymbol{y})$ is replaced by a Gaussian with $\boldsymbol{y}$-dependent mean $\bar{\boldsymbol{x}}(\boldsymbol{y})$ and covariance matrix $\{\Sigma_{ij}(\boldsymbol{y})\}$. The first representation provides simple expressions to eqs. (4), but the solution of eq. (11) is computationally more difficult to obtain; here we therefore present solutions based on the second representation

$$P(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{\sqrt{(2\pi)^K |\Sigma(\boldsymbol{y})|}} \exp\left[-\frac{1}{2}[\boldsymbol{x} - \bar{\boldsymbol{x}}(\boldsymbol{y})]^T \Sigma^{-1}(\boldsymbol{y})[\boldsymbol{x} - \bar{\boldsymbol{x}}(\boldsymbol{y})]\right] . \tag{12}$$

This results in the following dynamical equations for $\bar{\boldsymbol{x}}(\boldsymbol{y})$ and for $\Sigma_{ij}(\boldsymbol{y})$:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\bar{x}_i(\boldsymbol{y}) &= \frac{\eta}{\alpha}\bar{\mathcal{G}}_i(\boldsymbol{y}) + \eta\ [W\boldsymbol{y} + Y(\bar{\boldsymbol{x}}(\boldsymbol{y}) - R\boldsymbol{y})]_i\ , \\ \frac{\mathrm{d}}{\mathrm{d}t}\Sigma_{ij}(\boldsymbol{y}) &= \frac{1}{\alpha}\left[\eta\ (\bar{V}_{ij}(\boldsymbol{y}) + \bar{V}_{ji}(\boldsymbol{y}) - \bar{\mathcal{G}}_i(\boldsymbol{y})\bar{x}_j(\boldsymbol{y}) - \bar{\mathcal{G}}_j(\boldsymbol{y})\bar{x}_i(\boldsymbol{y})) + \eta^2\ \bar{Z}_{ij}(\boldsymbol{y})\right] + \\ &\quad +\eta\ [(S\Sigma(\boldsymbol{y}))_{ij} + (S\Sigma(\boldsymbol{y}))_{ji}] + \eta^2\ Z_{ij}\ , \end{aligned} \tag{13}$$

with the matrices $S = (V - WR^T)(Q - RR^T)^{-1}$ and $Y = (V - \langle \boldsymbol{\mathcal{G}}\bar{\boldsymbol{x}}^T\rangle)(Q - RR^T)^{-1}$, and with $\bar{\mathcal{G}}_i(\boldsymbol{y}) = \int \mathrm{d}\boldsymbol{x}\ \mathcal{G}_i(\boldsymbol{x}, \boldsymbol{y})\ P(\boldsymbol{x}|\boldsymbol{y})$, $\bar{V}_{ij}(\boldsymbol{y}) = \int \mathrm{d}\boldsymbol{x}\ \mathcal{G}_i x_j\ P(\boldsymbol{x}|\boldsymbol{y})$ and $\bar{Z}_{ij}(\boldsymbol{y}) = \int \mathrm{d}\boldsymbol{x}\ \mathcal{G}_i\mathcal{G}_j\ P(\boldsymbol{x}|\boldsymbol{y})$. Equations (13), (4) are solved numerically using the appropriate initial conditions, to predict the evolution of the macroscopic variables, and of the training and generalisation errors.

We validate our analysis by comparing the results obtained to those obtained from numerical simulations. We restrict our experiments to two cases: a) A realisable scenario where $K = M = 2$. b) An unrealisable scenario $K = 2, M = 3$. Teacher vectors in both cases are taken as orthogonal and of unit length. We use the following initial conditions for both theory and

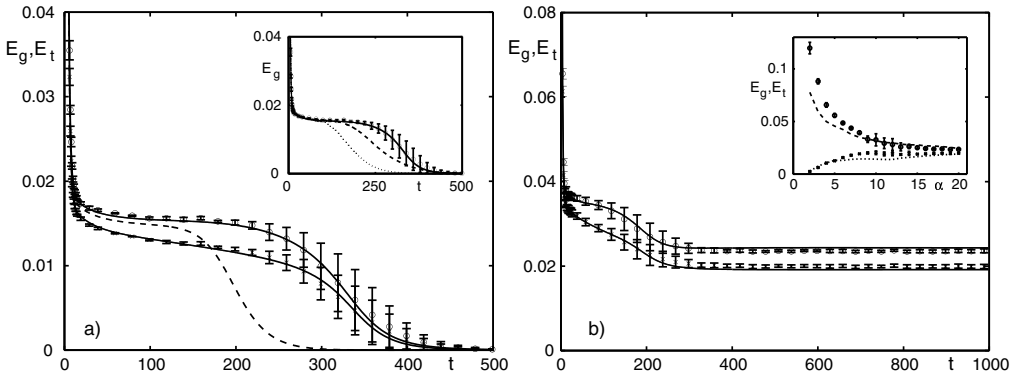


Fig. 2 – The evolution of the training and generalisation errors in comparison to those obtained from simulations for the case of $\alpha = 20$. (a) The theoretical values for the training (lower) and generalisation (higher) errors are represented by the solid lines; the training error simulation results for system size of $N = 5000$ are represented by symbols (mean values and error bars averaged over 10 trials, $\eta = 0.5$). Theoretical results for the infinite training set case have been added for comparison (dashed line). Inset: Finite-size effects are examined by plotting simulation results for the generalisation error for systems of size $N = 5000$ (circles with error bars), $N = 1000$ (dashed line) and $N = 500$ (dotted line); theoretical results are denoted by a solid line. (b) The learning dynamics in an unrealisable case ($K = 2, M = 3$) with initial conditions $R_{11}^0 = 0.05$, $Q_{11}^0 = 0.4$, $Q_{22}^0 = 0.6$, learning rate $\eta = 1$ and $\alpha = 20$; simulations were performed on a system of size $N = 1000$. Inset: The asymptotic ($t = 1000$) values of generalisation and training errors for different $\alpha$ values.

simulations: $Q_{11} = Q_{22} = 0.5$, $Q_{12} = Q_{21} = 0$, $R_{11} = 0.001$ and $R_{11} = 0.05$, $Q_{11} = 0.4$, $Q_{22} = 0.6$ for the two scenarios, respectively; other order parameters are set to zero. The initial joint probability $P[\boldsymbol{x}, \boldsymbol{y}]$ is assumed Gaussian, with the corresponding parameters. We first investigate the accuracy of our approximation in the case of low $\alpha$ values, where the accuracy of the approximation is expected to be the worst due to the (large-$\alpha$) approximation used. However, in these cases we will not be able to observe the breaking of the symmetric phase for computationally feasible system sizes. We will therefore concentrate on the prediction accuracy within the symmetric phase, where all vectors of the student system emulate the various vectors in the teacher system with equal success.

Figure 1 compares the numerical solutions of the analytical equations to simulation results obtained for various $\alpha$ values. The simulation results (denoted by symbols) were obtained for a system of size $N = 500$, initialised at random, while restricting the overlap values to those used in the analytical solutions. Figures 1a) and b) show the generalisation and training errors as functions of time, respectively. The insets show the evolution of the various overlaps for the case of $\alpha = 5$ compared to results obtained from simulations. The solutions obtained are in good agreement with the simulations, even at these low $\alpha$ values.

However, our main interest in the case of multilayer networks is in the symmetry-breaking process, whereby student vectors specialise, each learning to imitate a specific teacher vector. Note that an arbitrary choice of initial conditions, in the case of orthogonal teacher vectors of equal length and an infinite system size, would produce a divergence in the length of the symmetric phase, leading to a learning scenario of little interest, which cannot practically be validated against simulation results. Therefore, in spite of the fact that the analysis is carried out in the infinite system limit, we used initial conditions which correspond to a large (but not infinite) system size, which inevitably introduce a macroscopic symmetry breaking.

In fig. 2a) we show the evolution of both the generalisation and training errors for the

case of $\alpha = 20$ which is sufficiently high for observing the symmetry-breaking phenomena in simulations. The theoretical values for the training (lower) and generalisation (higher) errors are represented by the solid lines; the simulation results for system size of $N = 5000$ are represented by symbols (mean values and error bars) and were averaged over 10 trials. In the inset we examine the finite-size effects, comparing the theoretical results obtained for the generalisation error to the simulation results for $N = 500, 1000$ and 5000. Simulation results for lower $N$ values are represented by dashed ($N = 1000$) and dotted ($N = 500$) lines and were averaged over 30 trials. For brevity, only mean results are presented for the smaller $N$ values; error bars are generally similar to those of $N = 5000$. In fig. 2b) we examine the learning dynamics in the unrealisable case $K = 2, M = 3$; simulations were performed on a system of size $N = 1000$ (20 trials). The inset shows the asymptotic values of generalisation and training errors for different $\alpha$ values; these were calculated at $t = 1000$.

In summary, we have obtained a theoretical framework, based on the dynamical replica method, for the analysis of on-line learning scenarios in multi-layer networks where training examples are sampled with repetition from a fixed example set. To simplify the numerical solution we used the large-$\alpha$ approximation, which is highly suitable to the case examined, and solved the equations obtained using a parametrised approximated representation of the local fields conditional probability distribution $P(\boldsymbol{x}|\boldsymbol{y})$. The numerical results obtained are in good agreement with the simulations for both low and high $\alpha$ values. While we believe that the basic derivation provides an accurate description of the learning process, one may have to refine the approximations taken in the case of low $\alpha$ values, especially in scenarios where the parametrised probability distributions become inaccurate. The main drawback of the current framework is its complexity, which prevents one from obtaining fully analytical generic results; this will be the focus of future research in this area. One other natural extension of the current framework, now under way, is to include the possibility of noisy data and the use of simple regularisation methods which have been examined recently in the single-layer case [15]. Understanding the proper use of regularisation and early stopping methods, which has now come within reach using our analysis, will clearly be of great interest to practitioners.

* * *

DS and YX acknowledge support by EPSRC (GR/L52093) and the British Council (ARC1037).

REFERENCES

[1] CYBENKO G., *Math. Control Signals Systems*, **2** (1989) 303.
[2] BISHOP C. M., *Neural Networks for Pattern Recognition* (Oxford University Press, Oxford) 1995.
[3] MACE C. W. H. and COOLEN A. C. C., *Statistics Comput.*, **8** (1998) 55.
[4] SAAD D. (Editor), *On-line Learning in Neural Networks* (Cambridge University Press, Cambridge) 1998.
[5] KINZEL W. and RUJAN P., *Europhys. Lett.*, **13** (1990) 473.
[6] BIEHL M. and SCHWARZE H., *J. Phys. A*, **28** (1995) 643.
[7] SAAD D. and SOLLA S. A., *Phys. Rev. Lett.*, **74** (1995) 4337; *Phys. Rev. E*, **52** (1995) 4225.
[8] KROGH A. and HERTZ J. A., *J. Phys. A*, **25** (1992) 1135.
[9] HORNER H., *Z. Phys. B*, **86** (1992) 291; **87** (1992) 371.
[10] SOLLICH P. and BARBER D. in [4], p. 279.
[11] LÓPEZ B. and OPPER M., *Europhys. Lett.*, **49** (2000) 275.
[12] LEE S. and WONG K. Y. M., in *Advances in Neural Information Processing Systems*, Vol. **12**, edited by S. A. SOLLA, T. K. LEEN and K. MÜLLER (MIT Press, Cambridge, Mass.) 2000, p. 286.
[13] COOLEN A. C. C. and SAAD D.,in [4], p. 303; to be published in *Phys. Rev. E* (2000).
[14] COOLEN A. C. C., LAUGHTON S. N. and SHERRINGTON D., *Phys. Rev. B*, **53** (1996) 8184.
[15] MACE C. W. H. and COOLEN A. C. C. in [12], p. 237.