Premature Saturation in Backpropagation Networks: Mechanism and Necessary Conditions*

Javier E. Vitela and Jaques Reifman

Reactor Analysis Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439

*Work supported by the U.S. Department of Energy, Nuclear Energy Programs under contract W-31-109-ENG-38.

PREMATURE SATURATION IN BACKPROPAGATION NETWORKS: MECHANISM AND NECESSARY CONDITIONS

Abstract - *The mechanism that gives rise to the phenomenon of premature saturation of the output units of feedforward multilayer neural networks during training with the standard backpropagation algorithm is described. The entire process of premature saturation is characterized by three distinct stages and it is concluded that the momentum term plays the leading role in the occurrence of the phenomenon. The necessary conditions for the occurrence of premature saturation are presented and a new method is proposed, based on these conditions, that eliminates the occurrence of the phenomenon. Validity of the conditions and the proposed method are illustrated through simulation results.*

Keywords - feedforward neural networks, backpropagation training algorithm, premature saturation, flat spot, necessary conditions

1. INTRODUCTION

The slow convergence of the standard backpropagation (BP) algorithm for training feedforward multilayer neural networks (NN) is generally attributed to the fact that BP is a gradient descent-based method (Rumelhart et al., 1986). Yet, another reason for the slow convergence of BP training is the occurrence of the phenomenon of *premature saturation* (PS) of the network output units when the units are mapped by sigmoid-like functions (Franzini, 1987; Dahl, 1987; Fahlman, 1989; Chen and Mars, 1990; Lee et al., 1991; Balakrishnan and Honavar, 1992; Spartz and Honavar, 1993; Parekh et al., 1993; Vitela and Reifman, 1993). This undesirable phenomenon, sometimes referred in the literature as the *flat spot* problem, is characterized by the temporary trapping of the network output units at saturated activation levels during the early stages of the training process. While trapped, the saturated output units preclude any significant improvements in the training weights directly connected to these units, causing an unnecessary increase in the number of iterations required to train the network. The temporary trapping may occur for tens to thousands of iterations which can strongly affect the already slow convergence of the BP algorithm.

Although the PS problem has been widely recognized by researchers and NN users, to our knowledge, there is no work to date that correctly describes the dynamic *mechanism* that gives rise to the occurrence of the phenomenon. The purpose of this work is to analyze this mechanism, present the necessary conditions for its occurrence, and propose a modified BP algorithm that avoids PS.

In the next section, a brief description of the standard BP training algorithm and the characterization of the phenomenon of PS are presented, followed by a discussion in Sec. 3 of the mechanism that produces PS. In Sec. 4 the necessary conditions for the occurrence of PS are established and in Sec. 5 we describe the distinct characteristic stages of the phenomenon. In Sec. 6 we propose a new method, based on the necessary conditions, that prevents the occurrence of PS. Simulation results illustrating the validity of the necessary conditions and the proposed method are presented in Sec. 7, followed by a summary and conclusions in Sec. 8.

2. BACKPROPAGATION AND PREMATURE SATURATION

The BP algorithm trains a feedforward multilayer NN by iteratively searching for a set of weights **w** in weight-space that minimize the total training error E. Defining E (for reasons that will become clear later in the paper), as the sum of partial training errors E_j ($j=1,2,...,J_L$) associated with each one of the J_L output units of the last layer L of the network, we have

$$E = \sum_{j=1}^{J_L} E_j , \tag{1}$$

with

$$E_j = \frac{1}{2} \sum_{p=1}^{P} [t_{pj} - o_{pj}^{(L)}]^2 , \tag{2}$$

where t_{pj} and $o_{pj}^{(L)}$ are the desired target and the network actual activation level, respectively, for output unit j and pattern p ($p=1,2,...,P$). Each one of these partial training errors E_j associated with the corresponding output unit j, can be thought to constitute an individual error-surface $E_j(\mathbf{w})$ in weight-space.

Following Rumelhart's et al. (1986) notation, the activation level of the *n-th* unit in the *ℓ-th* layer for a given pattern p, $o_{pn}^{(\ell)}$, is given by

$$o_{pn}^{(\ell)} = f(net_{pn}^{(\ell)}) , \tag{3}$$

where $f(\cdot)$ is the activation function of the unit, assumed to be a sigmoid-like function, and $net_{pn}^{(\ell)}$ is the net input to the *n-th* unit. The net input is defined as

$$net_{pn}^{(\ell)} = \sum_{i=1}^{J_{\ell-1}} w_{ni}^{(\ell)} o_{pi}^{(\ell-1)} , \tag{4}$$

where $w_{ni}^{(\ell)}$ is the weight connecting the *i-th* unit in the *(ℓ-1)-th* layer with the *n-th* unit in the *ℓ-th* layer.

At the end of each iteration k, the weights $\mathbf{w}_k$ are updated through the weight update rule, $\mathbf{w}_{k+1} = \mathbf{w}_k + \Delta\mathbf{w}_k$, with

$$\Delta\mathbf{w}_k = -\eta\, \nabla E(\mathbf{w}_k) + \alpha\, \Delta\mathbf{w}_{k-1}, \tag{5}$$

where η and α are positive constants smaller than 1.0 known as the learning parameter and momentum parameter, respectively, and $\nabla E = \sum_{j=1}^{J_L} \nabla E_j$. The component of the gradient ∇E_j corresponding to weight $w_{ni}^{(\ell)}$ can be obtained recursively from

$$\frac{\partial E_j}{\partial w_{ni}^{(\ell)}} = \sum_{p=1}^{P} \delta_{pn}^{(\ell)} o_{pi}^{(\ell-1)}, \tag{6}$$

where $\delta_{pn}^{(\ell)}$ is associated with the training error E_j of the *j-th* output unit; and for sigmoid mapping activation functions it can be shown to be given by:

$$\delta_{pn}^{(\ell)} = \begin{cases} [o_{pn}^{(L)} - t_{pn}]\, o_{pn}^{(L)} [1 - o_{pn}^{(L)}]; & \text{for } \ell = L \text{ and } n = j, \\ 0; & \text{for } \ell = L \text{ and } n \neq j, \\ o_{pn}^{(\ell)} [o_{pn}^{(\ell)} - 1] \sum_{m=1}^{J_{\ell+1}} \delta_{pm}^{(\ell+1)}\, w_{mn}^{(\ell+1)}; & \text{for } 1 < \ell < L. \end{cases} \tag{7}$$

This expression is similar to the one obtained by Rumelhart et al. (1986), except that in this formulation it is explicitly shown that $\delta_{pn}^{(\ell)}$ is identically zero for weights that do not connect the units in the last hidden layer to the *j-th* output unit. Thus, when the training error E_j for a particular output unit j is backpropagated to the last hidden layer, it will only affect a subset $\mathbf{u}_j$ of the total weights connecting the J_{L-1} units of the last hidden layer and

the J_L output units. The affected weights $\mathbf{u}_j$ are those that are directly connected to output unit j, i.e., $\mathbf{u}_j = [w_{j1}^{(L)}, w_{j2}^{(L)}, \ldots\ldots, w_{jJ_{L-1}}^{(L)}]^T$.

The phenomenon of PS of the network output units is characterized by the fact that the activation level $o_{pj}^{(L)}$ of one or more output units approaches either 0 or 1 during the early stages of training for each training pattern p=1,2,...,P. As a consequence of the premature saturation of the output unit j, the factor $o_{pj}^{(L)}[1 - o_{pj}^{(L)}]$ in Eq. (7) corresponding to the slope of the sigmoid function approaches zero, causing the magnitude of the gradient components $\partial E_j / \partial w_{ji}^{(L)}$ for i=1, 2, ..., J_{L-1}, to attain small values. The weights $w_{ji}^{(L)}$ of the set $\mathbf{u}_j$ (which are all connected to the "saturated" output unit j), are then negligibly updated at each subsequent iteration causing both $\mathbf{u}_j$ and $o_{pj}^{(L)}$ to become *trapped* at their current values for a number of iterations. Such trapping of the weights $\mathbf{u}_j$ and the activation level $o_{pj}^{(L)}$ are generally characterized in the training error curve by regions of flat plateaus at relatively high error levels.

In the past, these plateaus have been erroneously interpreted by some authors as an intrinsic process used by NNs while constructing internal representations to distinguish between different input patterns. However, further research has shown that the only role of PS is to produce a detrimental effect in the training process, which is manifested by an increase in the number of training cycles required to release the trapped weights.

A number of researchers have addressed the problems posed by the PS of the network output units in order to accelerate convergence. In essence, the proposed approaches consider the modification of either the slope of the sigmoid function $o_{pj}^{(L)}[1 - o_{pj}^{(L)}]$ or the definition of the network training error E such that $\delta_{pn}^{(\ell)}$ in Eq. (7) remains finite even when an output unit is saturated. Franzini (1987) modified the standard BP algorithm by redefining the training error E in Eq. (1) such that $\delta_{pn}^{(\ell)}$ remains large when $o_{pj}^{(L)}[1 - o_{pj}^{(L)}]$ approaches zero and the absolute difference between the output and target values is near one, i.e., in the case of an output unit whose output is at the wrong end of the sigmoid.

Fahlman (1989) experimented with a family of learning algorithms that eliminates premature saturation by directly altering the derivative of the sigmoid such that it does not

go to zero. Of the modifications that he proposed, the one that seems to be most successful is the one in which a constant 0.10 was added to the value of $o_{pj}^{(L)}[1 - o_{pj}^{(L)}]$ before this value was used to scale the backpropagation error. Chen and Mars (1990), however, were not successful in applying this modified algorithm. According to their simulations, the change in the derivative of the sigmoid from $o_{pj}^{(L)}[1 - o_{pj}^{(L)}]$ to $o_{pj}^{(L)}[1 - o_{pj}^{(L)}]+0.10$ caused the weights to grow too fast, leading to floating point overflows during training. In order to circumvent premature saturation, they suggested the removal of $o_{pj}^{(L)}[1 - o_{pj}^{(L)}]$ from Eq. (7), i.e., setting the slope of the sigmoid equal to one, for output layer units and the usage of different learning parameters η for updating weights in different layers. Balakrishnan and Honavar (1992) handled the premature saturation problem by redefining the training error E as the mean squared error over the *inputs* to the output layer units rather than over the outputs as is conventionally done in BP. By redefining E and approximating the sigmoid by a straight line, $\delta_{pn}^{(\ell)}$ (for ℓ=L) becomes proportional to $1 / o_{pj}^{(L)}$ which permits the weights to be updated whenever there is an error in the output units. However, this method may lead to significant weight changes that result in large weights and oscillatory behavior.

A different approach due to Parekh et al. (1993) is based on an algorithm that works like BP unless the activation level of a unit in the output layer is greater than $1-\varepsilon$ while the target is zero, or if its activation level is less than ε and the target is one, where ε is a small positive constant. When that happens, the weights connecting all units in the last hidden layer to this unit in the output layer are updated through a predefined rule, such that the updated weights cause the activation level of this unit to fall in the $(\varepsilon, 1-\varepsilon)$ interval. Once the updated weights satisfy this requirement, the standard BP algorithm is then used to update all the weights of the network. Yet another simple modification of the standard BP algorithm has been proposed by Vitela and Reifman (1993), where the slope of the activation level is set to a constant value when the activation level of $o_{pj}^{(L)}$ falls in predefined saturation regions. The saturation regions are defined by values of $o_{pj}^{(L)}$ outside the (0.0025,0.9975) range, corresponding to slope values smaller than 1% of the maximum slope obtained at $o_{pj}^{(L)}$=0.5. The constant slope value of 0.09, adopted for the saturation

regions, corresponds to the slope value obtained when $o_{pj}^{(L)}=t_{pj}=0.9$ or 0.1, i.e., when the output units have reached their expected activation levels at the end of training.

The objective of these modifications to the BP algorithm is to reduce the training time by precluding the output units from getting stuck in the wrong state. The amount of improvement, as compared to the standard BP algorithm, varies for each approach and was found to be problem-dependent. In spite of the fact that these researchers have recognized the premature saturation problem, to our knowledge, only Lee et al. (1991) have attempted to explain the origin of this undesirable phenomenon. These authors analyzed the PS of the output units of the network as a *static* phenomenon that occurs at the first training cycle, as a consequence of the randomly chosen initial set of weights. They derived expressions that approximate the probability of the occurrence of PS at the first training cycle as a function of the value of the initial weights, the number of nodes in each layer, and the slope of the sigmoid function. In the next section, we describe a *dynamic* mechanism that produces the phenomenon of PS, followed by a set of necessary conditions that need to be satisfied for the phenomenon to occur.

3. THE MECHANISM THAT PRODUCES PREMATURE SATURATION

At the early stages of the training process, the activation levels of the output units $o_{pn}^{(L)}$ (n=1, 2, ..., J_L), are usually far away from their target values t_{pn} for most of the training patterns. In general, this causes the components of the weight update $\Delta \mathbf{w}_{k-1}$ in Eq. (5) belonging to vector $\mathbf{u}_n$ to attain values of the same order of magnitude as those of the components of $\mathbf{w}_{k-1}$. In this case, the updated weights $\mathbf{w}_k = \mathbf{w}_{k-1} + \Delta \mathbf{w}_{k-1}$, used during the pattern presentation at iteration k, may suddenly produce a large change in the net input $net_{pn}^{(L)}$ in Eq. (4) for some of the J_L output units, for all patterns p. This change can be such that it causes the activation levels $o_{pj}^{(L)}$ in Eq. (3), of a particular output unit j, to approach either 0.0 or 1.0 for all patterns, overshooting their target values t_{pj}.

To focus our discussion, let us assume that at iteration k, each training pattern p (p=1, 2,..., P) overshoots its associated target t_{pj} for output unit j, where t_{pj} is different from zero or one, and approaches 1.0. The following discussion, however, is general, and

similar reasoning applies to overshoots toward 0.0 and also when the target values are set to 0.0 and 1.0. Figure 1a illustrates this scenario for a given pattern p and associated target t_{pj}, where we plot delta $\delta_{pn}^{(\ell)}$ in Eq. (7), for ℓ=L and n=j, as a function of the activation level $o_{pj}^{(L)}$. Here, we assume that the unit's activation level o(k) at iteration k approaches 1.0 and is larger than its target value t_{pj} and the activation level o(k-1) at iteration k-1. Assuming that this scenario has occurred at the end of iteration k, the learning term $-\eta\nabla E(\mathbf{w}_k)$ and the momentum term $\alpha\Delta\mathbf{w}_{k-1}$ will tend to update the weights of $\mathbf{u}_j$ in opposite directions. The learning term will tend to update the weights such that at the next presentation of pattern p (iteration k+1), the activation level o(k+1) of the unit moves in the direction pointing toward its target t_{pj}, in an attempt to reduce the unit error E_j. On the other hand, since the momentum term represents the contribution of the previous weight update, it will tend to update the weights in the same direction as $\Delta\mathbf{w}_{k-1}$ did, i.e., in the direction such that o(k+1) is further away from t_{pj}, to the right of o(k), and closer to 1.0. The relative magnitude of the learning term and the momentum term will determine in which direction $\Delta\mathbf{w}_k$ will update the weights of $\mathbf{u}_j$.

In the case where the momentum term dominates, the new weight update $\Delta\mathbf{w}_k$ will produce a set of weights $\mathbf{w}_{k+1}$ which when presented with each pattern p will yield activation levels o(k+1) that are further away from their target and closer to 1.0 (see Fig. 1b). As a consequence, an increase in the partial training error E_j is observed at iteration k+1. If in addition, o(k) has already crossed the value o_{max} (or o_{min} if the unit is saturating to zero, see Fig. 1a), at which the magnitude of delta $\delta_{pj}^{(L)}$ has an extreme value, then $\delta_{pj}^{(L)}$ will decrease, being smaller in magnitude than its value at iteration k. A decrease in $\delta_{pj}^{(L)}$ can then lead to a decrease in the magnitude of $\partial E_j / \partial w_{ji}^{(L)}$ in Eq. (6). Although a decrease in $\delta_{pj}^{(L)}$ for all patterns at one iteration may not necessarily cause the magnitude of $\partial E_j / \partial w_{ji}^{(L)}$ to decrease (since it is composed by the summation of the product $\delta_{pj}^{(L)} o_{pi}^{(L-1)}$ over p), the magnitude of $\partial E_j / \partial w_{ji}^{(L)}$ will decrease with $\delta_{pj}^{(L)}$ if the changes in $o_{pi}^{(L-1)}$ cannot offset the decrease in $\delta_{pj}^{(L)}$.

At the end of iteration k+1, the diminished learning term will once again try to update the weights of $\mathbf{u}_j$ in a direction in which o(k+2) is closer to t_{pj}, while the momentum

term will try to update the weights in a direction in which o(k+2) tends to 1.0. In the case where the learning term can not offset the tendency of the momentum term, o(k+2) will become even closer to 1.0 (see Fig. 1c), the partial error E_j will increase, and the magnitude of the components of ∇E_j will decrease. This in turn, creates a self-propelling mechanism that reinforces itself at consecutive iterations causing, as an end result, the saturation of the activation levels of output unit j, $o_{pj}^{(L)}$, for all patterns p, to 1.0.

During the saturation process, the magnitude of the components of ∇E_j decrease at consecutive iterations as $\delta_{pj}^{(L)}$ decreases with $o_{pj}^{(L)}$ approaching 1.0. After just a few iterations into the saturation process, the magnitude of the components of ∇E_j become very small, approaching an asymptotic value of 0.0. With a negligible learning term, the weight update for the weights of $\mathbf{u}_j$ are solely based on the changes of the momentum term, i.e., $\Delta \mathbf{w}_k \approx \alpha \Delta \mathbf{w}_{k-1}$. Because the momentum parameter α is selected from the [0,1] interval, the contribution of the momentum term $\alpha \Delta \mathbf{w}$ in updating $\mathbf{u}_j$ also decreases at consecutive iterations, but at a slower rate than ∇E_j. As a consequence, the weight update $\Delta \mathbf{w}$ approaches 0.0, causing the weights $\mathbf{u}_j$ connecting all units of the last hidden layer to output unit j to become trapped at their current values, while the activation levels $o_{pj}^{(L)}$ remain at their saturated state of 1.0. Thus, the phenomenon of premature saturation precludes any significant changes in E_j (since the components of $\Delta \mathbf{w}$ corresponding to $\mathbf{u}_j$ approach 0.0), originating the characteristic flat plateau in the training error curve.

The temporary trapping of $\mathbf{u}_j$ and $o_{pj}^{(L)}$ may occur for tens to thousands of iterations, without contributing to the minimization of the total error E, until the recovery of the saturated unit. A saturated output unit starts recovering from its saturated state when $\mathbf{u}_j$ is updated in the negative gradient direction, i.e., a direction which minimizes E_j. Therefore, the recovery from saturation starts when the decreasing contribution of $\alpha \Delta \mathbf{w}_{k-1}$ to the weight update $\Delta \mathbf{w}_k$ has become small enough and is comparable to the negligible asymptotic contribution of $-\eta \nabla E(\mathbf{w}_k)$.

In general, PS is not manifested in all output units of the network. The occurrence of this phenomenon as well as the number of saturated units are strongly dependent on the

starting point in weight-space, the values of the parameters η and α, the topology of the network, the target values and the number of training patterns.

Although we identify the momentum term as the culprit for the occurrence of PS, it is possible to observe saturation without the use of the momentum term. For example, saturation can occur from the selection of either a large learning parameter or a large range of values for the initial weights (Lee et al., 1991). These cases are not related to the dynamic PS phenomenon that we discuss in this paper. Other phenomena, such as local minimum and saddle points, can also cause the occurrence of a flat plateau in the training error curve with or without a momentum term. However, these phenomena usually occur at later stages of the training process and can not be classified as PS since in their occurrence the activation levels $o_{pj}^{(L)}$ do not approach either 0 or 1 for each pattern p=1,2,...,P in the training set.

4. NECESSARY CONDITIONS FOR PREMATURE SATURATION

Based on the description of the mechanism that causes PS, we now present four necessary conditions that, if simultaneously satisfied, cause the saturation of output unit j.

To first order, the standard BP algorithm yields smaller values of the partial error $E_j(\mathbf{w}_{k+1})$ in Eq. (2) if the weight update $\Delta\mathbf{w}_k$ satisfies the inequality $\Delta\mathbf{w}_k \cdot \nabla E_j(\mathbf{w}_k) < 0$, which after substituting for $\Delta\mathbf{w}_k$ in Eq. (5), yields

$$\eta \nabla E(\mathbf{w}_k) \cdot \nabla E_j(\mathbf{w}_k) - \alpha \Delta\mathbf{w}_{k-1} \cdot \nabla E_j(\mathbf{w}_k) > 0. \tag{8}$$

However, at the onset of premature saturation, this inequality is not satisfied leading to the first two necessary conditions. The first necessary condition c1, reflects the fact that the projection of the momentum term points in the gradient direction ∇E_j, i.e.,

$$\Delta\mathbf{w}_{k-1} \cdot \nabla E_j(\mathbf{w}_k) > 0. \tag{c1}$$

This condition relates to our discussion in the previous section where the momentum term $\alpha\Delta\mathbf{w}_{k-1}$ tends to update the weights such that $o_{pj}^{(L)}$ approaches 1.0, rather than the target t_{pj}. The second condition c2,

$$\alpha\,|\Delta\mathbf{w}_{k-1}\cdot\nabla E_j(\mathbf{w}_k)| \;>\; \eta\,|\nabla E(\mathbf{w}_k)\cdot\nabla E_j(\mathbf{w}_k)|, \tag{c2}$$

reflects the fact that the magnitude of the projection of the momentum term should be larger than the projection of the learning term along the $\nabla E_j(\mathbf{w}_k)$ direction. This condition relates to the discussion in Sec. 3 where the momentum term dominates the direction of change in the weight update $\Delta\mathbf{w}_k$. These first two conditions highlight the role of the momentum term $\alpha\Delta\mathbf{w}_{k-1}$ in Eq. (5) as the driving force that governs the updating of the weights at the onset of PS.

The third condition c3,

$$|\nabla E_j(\mathbf{w}_{k+1})| < |\nabla E_j(\mathbf{w}_k)|, \tag{c3}$$

requires the magnitude of the gradient of E_j to decrease at successive iterations, as described in the previous section, allowing the momentum term to drive the weight update.

The fourth and last condition c4, requires that for all patterns p (p=1, 2, ..., P),

$$\begin{aligned} &\text{either,} \quad o_{pj}^{(L)}(\mathbf{w}_{k+1}) \;>\; o_{max}\,, \qquad \text{for saturation to 1.0, or} \\ &\qquad\qquad o_{pj}^{(L)}(\mathbf{w}_{k+1}) \;<\; o_{min}\,, \qquad \text{for saturation to 0.0,} \end{aligned} \tag{c4}$$

where

$$o_{max} = \frac{1}{3}\left(1 + t_{pj} + \sqrt{1 - t_{pj} + t_{pj}^2}\right), \text{ and} \tag{9a}$$

$$o_{min} = \frac{1}{3}\left(1 + t_{pj} - \sqrt{1 - t_{pj} + t_{pj}^2}\right). \tag{9b}$$

are the activation levels corresponding to the maximum and minimum values of delta $\delta_{pj}^{(L)}$, respectively, given the target value t_{pj} (see Fig. 1a). If satisfied, condition c4 along with condition c3, guarantee that the decrease in ∇E_j is due to the fact that output unit j is approaching saturation, rather than t_{pj}.

For the onset of saturation to occur, this set of four necessary conditions must be simultaneously satisfied and usually remain satisfied for a number of consecutive iterations. These conditions are non-local, that is, they depend on calculations performed at three consecutive iterations, k-1, k, and k+1, which, if satisfied, characterize the onset of PS at iteration k. However, as the process of PS evolves beyond the onset of saturation, each one of the four conditions is no longer satisfied, although the saturation process has not yet ended. Depending on which conditions are satisfied after the onset of saturation, different stages of the PS process can be defined. The definition of the stages of PS and associated requirements in the four necessary conditions is discussed in the next section.

5. STAGES OF THE PROCESS OF PREMATURE SATURATION

With the help of the necessary conditions, the analysis presented in the foregoing sections allows us to partition the entire process of PS into three distinct stages: beginning of saturation, saturation plateau, and complete recovery from saturation. The first stage, beginning of saturation, corresponds to the iterations of the BP algorithm where conditions c1 through c4 are simultaneously satisfied. During this stage, the self-propelling PS mechanism is launched, causing $|\nabla E_j|$ to decrease monotonically and to approach an asymptotic value of zero. The absolute value of the gradient $|\nabla E_j|$ at the end of this first stage, i.e., at the last iteration for which the four conditions are satisfied, denotes the extent of saturation of the output unit, with smaller values indicating higher degrees of saturation and vice versa.

The second stage, saturation plateau, starts at the first iteration after the beginning of saturation stage where any one of the three conditions, c1-c3, is no longer satisfied. Later in this stage, however, it is possible that these three conditions may once again

become simultaneously satisfied. On the other hand, condition c4 remains satisfied throughout this stage.

When the degree of saturation at the beginning of the second stage is high, both the momentum term and the learning term are small and produce negligible updates in the weights $\mathbf{u}_j$ at a single iteration. The trapping of the weights $\mathbf{u}_j$ at their current values causes the activation of the output unit to remain saturated, which, in turn, precludes changes in the error E_j. This is generally reflected as a saturation plateau in the training curve, i.e., a plot of the total training error E versus iteration number, at relatively large values of E. The length of the saturation plateau is strongly dependent on the degree of saturation reached at the end of the first stage, with long lengths associated with high degrees of saturation and vice versa. In some cases, the degree of saturation in not high enough and the unit shows only slight tendency to saturate, in which case the saturation plateau may be very short or not noticeable at all.

The recovery from saturation starts at some point during this second stage, when the learning term and the momentum term systematically produce, at each iteration, weight updates that decrease E_j. Because of the negligible value of the weight updates during this stage of the PS process, only cumulative changes over a large number of iterations can untrap the weights $\mathbf{u}_j$ and lead to the recovery process. For this reason, the second stage is generally the longest of the three stages.

The third and last stage, the complete recovery from saturation, starts at the first iteration after the saturation plateau stage where condition c4 is no longer satisfied. This generally occurs after $|\nabla E_j|$ has increased significantly, suddenly untrapping the weights $\mathbf{u}_j$ and reversing the self-propelling effects of PS of the output unit described in Sec. 3. This last stage ends when the value of $|\nabla E_j|$ reaches a maximum, followed by a quick settle down, allowing the NN to resume a normal training process.

6. PREVENTING PREMATURE SATURATION

In this section we propose a new method that prevents the occurrence of PS. Unlike the *ad hoc* methods discussed in Sec. 2 that consider arbitrary modifications of

either the slope of the sigmoid function or the definition of the training error E such that $\delta_{pn}^{(\ell)}$ in Eq. (7) remains finite even when an output unit is saturated, the proposed method is based on the mechanism that causes PS and takes into account the necessary conditions presented in Sec. 5.

Since for the onset of saturation to occur the four necessary conditions c1-c4 must be simultaneously satisfied and remain satisfied for a number of consecutive iterations, we can prevent PS by modifying the training parameters, such that at least one of the conditions is not satisfied. Analysis of the four conditions indicates that only condition c2 contains training parameters, α and η, that can be temporarily modified to preclude the formation of the beginning of the saturation stage and prevent saturation of an output unit. In the new approach, we propose to temporarily modify α. Once c1-c4 are satisfied at iteration k, a value for the momentum parameter α is calculated such that condition c2 is not satisfied at iteration k+1. The calculated value of α is then used to compute the weight update $\Delta \mathbf{w}_{k+1}$ in Eq. (5). If more than one output unit satisfies the four conditions, α is calculated for each one of those units and the smallest α is used to update $\Delta \mathbf{w}_{k+1}$. After iteration k+1, the original value of α is used again as long as c1-c4 are not simultaneously satisfied. That is, the algorithm works like the standard BP unless the four conditions are satisfied. By temporarily modifying α we prevent the momentum term from dominating the direction of change in $\Delta \mathbf{w}_{k+1}$ when its tendency is to update the weights such that $o_{pj}^{(L)}$ approaches 0.0 or 1.0, rather than the target t_{pj}.

Analyses of the *ad hoc* methods discussed in Sec. 2 indicate that the modifications proposed in those methods also prevent the four necessary conditions from being simultaneously satisfied. For example, the modification of the slope of the sigmoid function proposed by Fahlman (1989) and Vitela and Reifman (1993) affects condition c3. By arbitrarily increasing the value of the slope when the slope is small, these methods cause an increase in the magnitude of the gradient, preventing condition c3 from being satisfied.

Although the proposed method follows the mathematical formulation of the four necessary conditions to catch the onset of PS and prevent its occurrence, the method does

not necessarily catch PS earlier or guarantee a faster convergence than the *ad hoc* methods. The methods that modify the slope of the sigmoid function may catch the onset of PS by observing that the unit activation has saturated, (based on a prespecified threshold value), while the training error is still large. Depending on the value used for the threshold, these methods can detect the onset of PS before or after the first iteration for which conditions c1-c4 are simultaneously satisfied. Which method detects the onset of PS the earliest is not important because the units saturate very quickly and the different methods detect the onset of PS within a few iterations of each other. Since the proposed method and the slope-based methods may detect the onset of saturation at distinct iterations, and, when detected, each method modifies the BP algorithm differently, the updated weights will be different, leading to different trajectories in weight-space in subsequent updates. Therefore, the number of iterations to converge for each approach is case-dependent, and a determination of which of the two approaches will converge faster cannot be made a priori. The proposed method, however, is systematic and is based on the causes, rather than the consequences, of PS.

7. SIMULATION RESULTS

In order to illustrate the validity of the necessary conditions described in Sec. 4 and the proposed modification of the BP algorithm discussed in the previous section, we present three cases from two distinct classification problems in which PS occurs during the training sessions.

Case 1: The first case comes from a training session for the classification of three component failures in a nuclear power plant (Reifman and Vitela, 1994). The network consisted of three layers with 20-20-3 units per layer, respectively, where the desired target values t_{pj} for the three output units were set to either 0.1 or 0.9, depending on the training pattern. The value of the learning parameter η was fixed at 0.1 throughout the training session, and the value of the momentum parameter α was set to zero for the first two iterations, and after that it was set to 0.9. An iteration in the BP algorithm consisted of the

presentation of the entire set of 108 training patterns, corresponding to 36 patterns for each one of the three component failures, after which the weights were updated.

Figure 2 shows the behavior of the total training error E for a training session in which a set of randomly selected weights caused two units, out of the three output units, to saturate prematurely. The occurrence of the PS phenomenon is clearly represented in the figure by regions of flat plateaus at high error levels. The corresponding training errors E_j, for j=1,2,3, associated with each one of the three output units is illustrated in Fig. 3. This figure shows that output units 1 and 3 are the two saturated units responsible for the formation of the flat plateaus in Fig. 2 and that the sharp decrease in the total training error E around iteration 350 is caused by the recovery from saturation of output unit 3.

During the beginning of saturation stage of the phenomenon of PS, the four necessary conditions defined in Sec. 4 are simultaneously satisfied. As illustrated in Fig. 3, the training errors E_1 and E_3 increase after a couple of iterations of the algorithm. The increase in E_1 and E_3 is a consequence of the satisfaction of necessary conditions c1 and c2, and expresses the fact that the weights directly connected to output units 1 and 3 are not being updated in a minimizing direction. Necessary condition c3, which requires that the magnitude of the gradient $|\nabla E_j|$ for saturating unit j decrease monotonically at successive iterations, is also satisfied after the second iteration for both output units as illustrated in Fig. 4.

Table I summarizes the results of testing output unit 1 for the four necessary conditions during the iterations in which PS occurs. Each entry in columns two through five indicates the status of the four necessary conditions. A necessary condition is satisfied if the associated entry is assigned the letter "Y", and is not satisfied if it is assigned the letter "N". Column six presents the magnitude of the gradient squared $|\nabla E_1|^2$, while the last column indicates the stage, e.g., beginning of saturation, saturation plateau, and complete recovery from saturation, of PS for unit 1 as a function of the iterations.

Table I shows that the four necessary conditions are simultaneously satisfied between iterations 3 and 14, constituting the first or the beginning of saturation stage. The second stage starts at iteration 15 where condition c2 is no longer satisfied, and ends at

iteration 28558 which is the last iteration where condition c4 is satisfied. The second stage is generally the longest of the three stages. The recovery from saturation of unit 1 begins around iteration 6000 where $|\nabla E_1|$ starts to increase monotonically, leading to the complete recovery from saturation which starts at iteration 28559. This last stage lasts only a few iterations until iteration 28563, after which there is a sudden decrease in $|\nabla E_1|^2$, indicating that unit 1 has fully recovered and can resume its normal approach toward the minimum of the total training error. A typical behavior of the activation levels of the three output units corresponding to an arbitrary teaching pattern with targets $t_{p1}=0.9$, $t_{p2}=0.1$, and $t_{p3}=0.1$, is illustrated in Fig. 5. This figure illustrates the fact that both units 1 and 3 saturate to zero during PS and that saturation occurs very quickly.

Table II shows the results of testing output unit 3 for the four necessary conditions during the iterations in which PS occurs. Like in unit 1, the beginning of saturation occurs between iterations 3 and 14, and the saturation plateau stage starts at iteration 15 where condition c2 is no longer satisfied. This second stage ends at iteration 353 since condition c4 is not satisfied at iteration 354. The length of the saturation plateau for this unit is much shorter than for unit 1, as it is a function of the degree of saturation reached by the unit at the end of the first stage. The higher degree of saturation of unit 1 (see Fig. 4) causes the unit to delay its recovery from its saturated state, which is manifested by a longer length of the saturation plateau in Fig. 3. The recovery from saturation of unit 3 begins around iteration 70, leading to the complete recovery at iteration 354, which ends at iteration 356 where the magnitude $|\nabla E_3|$ reaches a maximum value.

To compare these results of the BP algorithm with the proposed approach discussed in Sec. 6 and the ad hoc slope modification approach developed by Vitela and Reifman (1993), the network was retrained twice with these algorithms. In the retraining, the same network topology, initial set of weights, and training parameters were used. Both of these methods work like BP unless the onset of saturation is detected. In the proposed method, the onset of saturation is detected when conditions c1-c4 are simultaneously satisfied, and in the slope modification method when the activation level for any one pattern falls outside the prespecified range (0.0025,0.9975), in which case the slope of the sigmoid is

arbitrarily set to 0.09 for those patterns. By arbitrarily increasing the slope of the sigmoid, the method of Vitela and Reifman causes an increase in the magnitude of the gradient, preventing condition c3 from being satisfied.

Figures 6 and 7 show the total training error E and the partial errors E_j, for j=1,2,3, obtained by training the network with the proposed approach and the slope modification approach (Vitela and Reifman, 1993), respectively. Both methods prevent PS and converge much faster than the standard BP algorithm (see Figs. 2 and 3). In this case, the convergence criterion $|t_{pj} - o_{pj}^{(L)}| < 0.03$ (for all patterns p=1,2,...,P and output units j=1,2,3) was used. As indicated in the first column of Table III, the standard BP algorithm converged in 48392 iterations, the slope modification approach converged in 28312 iterations, and the proposed approach converged in 16730 iterations. When saturation occurs, both modified BP algorithms will generally converge faster than the standard BP. In some cases the proposed approach will converge faster than the slope modification and in other cases the slope modification will converge faster.

For the proposed approach, the onset of saturation was detected at the third iteration when the necessary conditions were first satisfied for units 1 and 3 (see Tables I and II) and a new value of the momentum parameter α was calculated and used to update the weights. At the fourth iteration condition c2 was not satisfied for either unit due to the modified value of α. But at the fifth iteration, the conditions were satisfied for unit 2. A new value of α was calculated and the standard BP algorithm was resumed at the sixth iteration until the end of training. For the slope modification approach, the onset of saturation was also detected at the third iteration when the activation level for at least one of the patterns for output unit 3 became smaller than the lower bound 0.0025 (see Fig. 5). The slope of the sigmoid was modified to 0.09 at the third iteration and remained modified until the sixth iteration. At the seventh iteration, the activation of each output unit for each pattern was larger than 0.0025 and the standard BP was resumed until the end of training. By changing the slope of the sigmoid function, necessary condition c3 was not satisfied at the fourth iteration. In spite of the small number of iterations for which the different algorithms are active, they produced significantly different weights because of the

comparable magnitude of Δw_k and w_k at the early stages of training and the different contributions of the learning term and momentum term in updating Δw_k in the two methods.

Case 2: The second example is taken from the same problem as in case 1. We used the same network topology and same learning and momentum parameters, but here we started the training from a different position in weight-space. Figure 8 shows the behavior of the total training error E for the first one-hundred iterations where no obvious signs of PS are observed. A closer analysis, however, indicates that the four conditions for PS are simultaneously satisfied between iterations 3 and 6 for output unit 2 (see Table IV). The saturation plateau stage occurs between iterations 7 and 22, and the complete recovery stage takes place between iterations 23 and 28.

Figure 9 shows the behavior of each one of the three training errors E_j associated with the three output units, where the saturation of unit 2 becomes apparent. Figure 10 shows the behavior of $|\nabla E_j|$, and indicates the very low degree of saturation of unit 2 at the end of the beginning of saturation stage (iteration 6). The low degree of saturation is reflected by a fast recovery from saturation of this unit which makes the PS phenomenon unnoticeable in Fig. 8. During this ephemeral saturation, unit 2 saturates to zero, as can be observed from a typical plot of the activation level of the three output units shown in Fig. 11, for an arbitrary pattern.

This case was also rerun with the two modified BP algorithms. Both methods prevented the occurrence of PS and converged faster than the standard BP. The same convergence criterion applied in Case 1 was used here. As indicated in the second column of Table III, the standard BP algorithm converged in 21941 iterations, the slope modification approach converged in 15661 iterations, and the proposed approach converged in 18638 iterations. In the slope modification approach, the onset of saturation was detected at the fourth iteration when the slope of sigmoid function was set to 0.09, causing condition c3 not to be satisfied for the fifth iteration. The standard BP was resumed at the sixth iteration. In the proposed approach, the onset of saturation was

detected at the third iteration (see Table IV), after which the standard BP algorithm was resumed.

Case 3: The third case comes from a training session for the classification of welded nuclear fuel elements (Reifman et al., 1995). Features extracted from digitized images of end plugs welded onto the top of reprocessed nuclear fuel elements were used to classify the weld as acceptable or unacceptable using a feedforward NN. The network consisted of three layers with 19-15-1 units per layer, respectively, where the target value of the output unit t_p was set to 0.1 or 0.9 depending on the training pattern p. As in the previous cases, the weights were updated after the presentation of the entire training set, which consisted of 70 patterns, with the first 44 corresponding to acceptable welds and the remaining 26 associated with unacceptable welds.

Figure 12 shows the behavior of the total training error $E=E_1$ for three different values, 0.9, 0.85, and 0.8, of the momentum parameter α. The same learning parameter, $\eta=0.1$, and the same initial set of weights were used in all three training sessions. As observed from the figure, the two training sessions using larger values for α, 0.9 and 0.85, saturated prematurely, while the phenomenon was not observed for the smaller $\alpha=0.8$. Tables V and VI present the results of testing the four necessary conditions for PS and the associated saturation stages for the training sessions with α set to 0.9 and 0.85, respectively. The first stage is shorter for $\alpha=0.85$, which highlights the role of the momentum term in the occurrence of PS. The contribution of the momentum term in updating the weights toward saturation, viz. condition c2, will diminish faster for smaller values of the momentum parameter. When α, and thus the magnitude of the momentum term, becomes smaller, PS is not observed, as is the case for $\alpha=0.8$. PS was also not observed in other simulation runs of this classification problem when we set $\alpha<0.8$.

Figure 13 illustrates the variations of the magnitude of ∇E for the three simulation runs, and Fig. 14 shows the typical behavior of the activation level for the same arbitrary teaching pattern with target $t_p=0.1$. Unlike the other two previous cases, Fig. 14 shows

that in this case the output unit prematurely saturates to 1, for α set to 0.9 and 0.85, instead of zero.

This case was also rerun with the two modified BP algorithms for α=0.9 and α=0.85. For each value of α, both methods prevented the occurrence of PS and converged faster than the standard BP. In this case, the convergence criterion $E < 0.03$ was used. The number of iterations taken by each algorithm to converge is illustrated in the last two columns of Table III. Figure 15 shows the training error curves for the slope modification approach and the proposed approach for α=0.85. The two curves show similar behavior which is in great contrast to the results obtained with the standard BP algorithm (see Fig. 12). Flat plateaus are not observed and the number of iterations to converge is reduced by a factor greater than 3.0. Similar results were obtained for α =0.90.

8. SUMMARY AND CONCLUSIONS

In this work we analyze the dynamic mechanism by which the phenomenon of PS of the output units is produced during training with the standard BP algorithm. In addition, a set of four necessary conditions that should be satisfied when a given output unit is to saturate prematurely is established, and it is concluded that the momentum term plays the leading role in the occurrence of the phenomenon. A modified BP algorithm that uses different values for the momentum parameter when the necessary conditions are satisfied is also proposed for avoiding PS. Our analysis also shows that the entire PS process can be partitioned into three distinct stages, beginning of saturation, saturation plateau, and the complete recovery from saturation, depending on which of the four conditions are satisfied.

The validity of these results is illustrated through three examples where PS was encountered during the training sessions of a feedforward multilayer network. Additional experiments performed by changing the values of the learning parameter and momentum parameter, as well as running the same problem with different sets of initial weights, confirmed the results presented here.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy, Nuclear Energy Program, under contract W-31-109-ENG-38.

DISCLAIMER

REFERENCES

Balakrishnan, K., and Honavar, V. (1992). Improving Convergence of Back-Propagation by Handling Flat-Spots in the Output Layer. In I. Aleksander and J. Taylor (Eds.), *Proceedings of the International Conference on Artificial Neural Networks* (vol. 2, pp. 1003-1009). Elsevier Science.

Becker, S., and Le Cun, Y. (1988). Improving the Convergence of Back-Propagation Learning with Second Order Methods. In D. Touretzky, G. Hinton, and T. Sejnowski (Eds.), *Proceedings of the Connectionist Models Summer School* (pp. 29-37). San Mateo, CA: M. Kaufmann.

Chen, J. R., and Mars, P. (1990). Stepsize Variation Methods for Accelerating the Back-Propagation Algorithm. In M. Caudill (Ed.), *Proceedings of the International Joint Conference on Neural Networks* (vol. I, pp. 601-604). Piscataway, NJ: IEEE Neural Networks Council.

Dahl, E. D. (1987). Accelerated Learning Using the Generalized Delta Rule. In M. Caudill and C. B. Butler (Eds.), *Proceedings of the International Joint Conference on Neural Networks* (vol. II, pp. 523-530). Piscataway, NJ: SOS Printing.

Fahlman, S. E. (1989). Faster-Learning Variations on Back-Propagation: An Empirical Study. In D. Touretzky, G. Hinton, and T. Sejnowski (Eds.), *Proceedings of the Connectionist Models Summer School* (pp. 38-51). San Mateo, CA: M. Kaufmann.

Franzini, M. A. (1987). Speech Recognition With Back Propagation. *Proceedings of the Ninth Annual Conference of the IEEE Engineering in Medicine and Biology Society* (vol. 3, pp. 1702-1703).

Lee, Y., Oh, S. H., and Kim, M. W. (1991). The Effect of Initial Weights on Premature Saturation in Back-Propagation Training. In Staff (Eds.), *Proceedings of the International Joint Conference on Neural Networks* (vol. I, pp. 765-770). Piscataway. NJ: IEEE Neural Networks Council.

Parekh, R., Balakrishnan, K., and Honavar, V. (1993). An Empirical Comparison of Flat-Spot Elimination Techniques in Back-Propagation Networks. *Proceedings of the Third Workshop on Neural Networks: Academic/Industrial/NASA/Defense* (pp. 55-60). San Diego, CA: Society for Computer Simulation.

Reifman, J. and Vitela, J. E. (1994). Accelerating Learning of Neural Networks with Conjugate Gradients for Nuclear Power Plant Applications. *Nuclear Technology* (vol. 106, pp. 225-241).

Reifman, J., Gibbs, K. S., Vitela, J. E., and Benedict, R. W. (1995). Automatic Inspection for Remotely Manufactured Fuel Elements. *Proceedings of the American Nuclear Society 6th Topical Meeting on Robotics and Remote Systems* (vol. 2, pp. 696-703). La Grange Park, IL: American Nuclear Society.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning Internal Representations by Error Propagation. In D. E. Rumelhart and J. C. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (vol. I, pp. 319-362). Cambridge, MA: MIT Press.

Spartz, R. and Honavar, V. (1993). An Empirical Analysis of the Expected Source Values Rule. *Proceedings of the Third Workshop on Neural Networks: Academic/Industrial/NASA/ Defense* (pp. 95-100). San Diego, CA: Society for Computer Simulation.

Vitela, J., and Reifman, J. (1993). Enhanced Backpropagation Training Algorithm for Transient Event Identification. *Transactions of the American Nuclear Society* (vol. 69, pp. 148-149). La Grange Park, IL: American Nuclear Society.

Table I. Results of the Four Necessary Conditions for Output Unit 1 in Case 1

Iteration Number	Condition c1	Condition c2	Condition c3	Condition c4	$\|\nabla E_1\|^2$	Stage of Saturation
1	N	N	N	N	0.48E+02	
2	N	N	Y	Y	0.33E+02	
3	Y	Y	Y	Y	0.81E+01	
4	Y	Y	Y	Y	0.10E+00	
5	Y	Y	Y	Y	0.26E-02	
.	.	.	.	.	.	Beginning
.	.	.	.	.	.	of
						Saturation
12	Y	Y	Y	Y	0.22E-07	
13	Y	Y	Y	Y	0.68E-08	
14	Y	Y	Y	Y	0.18E-08	
15	Y	N	Y	Y	0.11E-08	
.	.	.	.	.	.	
.	.	.	.	.	.	Saturation
						Plateau
28557	N	Y	N	Y	0.40E+00	
28558	N	Y	N	Y	0.94E+00	
28559	N	Y	N	N	0.28E+01	
28560	N	Y	N	N	0.13E+02	Complete
28561	N	N	N	N	0.70E+02	Recovery
28562	Y	N	N	N	0.97E+02	
28563	Y	N	Y	N	0.12E+03	
28564	N	Y	N	N	0.82E+00	
.	.	.	.	.	.	
.	.	.	.	.	.	

Table II. Results of the Four Necessary Conditions for Output Unit 3 in Case 1

Iteration Number	Condition c1	Condition c2	Condition c3	Condition c4	$\|\nabla E_3\|^2$	Stage of Saturation
1	N	N	N	N	0.11E+03	
2	N	N	Y	N	0.25E+03	
3	Y	Y	Y	Y	0.12E+02	
4	Y	Y	Y	Y	0.19E+00	
5	Y	Y	Y	Y	0.21E-01	
						Beginning
.	.	.	.	.	.	of
.	.	.	.	.	.	Saturation
12	Y	Y	Y	Y	0.23E-04	
13	Y	Y	Y	Y	0.15E-04	
14	Y	Y	Y	Y	0.96E-05	
15	Y	N	Y	Y	0.78E-05	
.	.	.	.	.	.	
.	.	.	.	.	.	Saturation
						Plateau
352	N	Y	N	Y	0.12E+01	
353	N	Y	N	Y	0.44E+01	
354	N	N	N	N	0.25E+02	Complete
355	N	N	N	N	0.85E+02	Recovery
356	Y	N	Y	N	0.32E+03	
357	N	Y	Y	N	0.70E+01	
.	.	.	.	.	.	
.	.	.	.	.	.	

Table III. Comparison on the Number of Iterations to Converge for the Backpropagation Algorithm, the Slope Modification Algorithm, and the Proposed Algorithm

	Case 1	Case 2	Case 3	
			α=0.90	α=0.85
Backpropagation	48,392	21,941	3,120	7,785
Slope Modification	28,312	15,661	1,347	2,069
Proposed Approach	16,730	18,638	1,693	2,323

Table IV. Results of the Four Necessary Conditions for Output Unit 2 in Case 2

Iteration Number	Condition c1	Condition c2	Condition c3	Condition c4	$\|\nabla E_2\|^2$	Stage of Saturation
1	N	N	N	N	0.34E+02	
2	N	N	Y	N	0.22E+02	
3	Y	Y	Y	Y	0.10E+02	Beginning
4	Y	Y	Y	Y	0.30E+00	of
5	Y	Y	Y	Y	0.25E-01	Saturation
6	Y	Y	Y	Y	0.58E-02	
7	Y	N	N	Y	0.28E-02	
8	N	Y	N	Y	0.75E-02	
.	.	.	.	.	.	Saturation
.	.	.	.	.	.	Plateau
21	N	Y	N	Y	0.73E+00	
22	N	Y	N	Y	0.12E+01	
23	N	Y	N	N	0.21E+01	
24	N	Y	N	N	0.47E+01	
25	N	Y	N	N	0.15E+02	Complete
26	N	Y	N	N	0.25E+02	Recovery
27	Y	Y	N	N	0.70E+02	
28	Y	N	Y	N	0.12E+03	
29	Y	N	N	N	0.46E+01	
.	.	.	.	.	.	
.	.	.	.	.	.	

Table V. Results of the Four Necessary Conditions for the Output Unit With the Momentum Parameter set to 0.90 in Case 3

Iteration Number	Condition c1	Condition c2	Condition c3	Condition c4	$\|\nabla E_1\|^2$	Stage of Saturation
.	.	.	.	.	.	
9	Y	N	N	N	0.12E+03	
10	N	N	Y	N	0.18E+03	
11	Y	Y	Y	Y	0.24E+02	
12	Y	Y	Y	Y	0.61E+00	
.	.	.	.	.	.	Beginning
.	.	.	.	.	.	of
						Saturation
69	Y	Y	Y	Y	0.47E-07	
70	Y	Y	Y	Y	0.46E-07	
71	Y	Y	N	Y	0.45E-07	
72	Y	N	N	Y	0.45E-07	
.	.	.	.	.	.	Saturation
.	.	.	.	.	.	Plateau
1641	N	Y	N	Y	0.38E+00	
1642	N	Y	N	Y	0.71E+00	
1643	N	Y	N	N	0.15E+01	
1644	N	Y	N	N	0.37E+01	
1645	N	Y	N	N	0.99E+01	Complete
1646	N	Y	N	N	0.11E+02	Recovery
1647	Y	Y	N	N	0.19E+02	
1648	Y	N	Y	N	0.48E+02	
1649	N	N	Y	N	0.33E+02	
.	.	.	.	.	.	
.	.	.	.	.	.	

Table VI. Results of the Four Necessary Conditions for the Output Unit With the Momentum Parameter set to 0.85 in Case 3

Iteration Number	Condition c1	Condition c2	Condition c3	Condition c4	$\|\nabla E_1\|^2$	Stage of Saturation
.	.	.	.	.	.	
8	Y	N	Y	N	0.18E+03	
9	N	N	Y	Y	0.10E+03	
10	Y	Y	Y	Y	0.97E+01	
11	Y	Y	Y	Y	0.17E+00	
.	.	.	.	.	.	Beginning
.	.	.	.	.	.	of
						Saturation
63	Y	Y	Y	Y	0.63E-08	
64	Y	Y	Y	Y	0.62E-08	
65	Y	N	N	Y	0.62E-08	
66	N	N	N	Y	0.62E-08	
.	.	.	.	.	.	Saturation
.	.	.	.	.	.	Plateau
5986	N	Y	N	Y	0.44E+00	
5987	N	Y	N	Y	0.82E+00	
5988	N	Y	N	N	0.18E+01	
5989	N	Y	N	N	0.43E+01	
5990	N	Y	Y	N	0.11E+02	Complete
5991	N	Y	N	N	0.12E+02	Recovery
5992	Y	Y	N	N	0.25E+02	
5993	Y	N	Y	N	0.49E+02	
5994	N	N	Y	N	0.17E+02	
.	.	.	.	.	.	
.	.	.	.	.	.	

LIST OF FIGURES

Fig. 10. Behavior of the Magnitude of the Gradient for the Partial Training Errors in Case 2 Reflecting the Effects of Premature Saturation in Output Unit 2

Fig. 11. Typical Activation Levels of the Three Output Units for a Given Pattern in Case 2 When Output Unit 2 Prematurely Saturate to zero

Fig. 12. Behavior of the Total Training Error for Three Simulation Runs in Case 3 With α set to 0.90, 0.85, and 0.80, Respectively, Reflecting the Effects of Premature Saturation for α values of 0.90 and 0.85

Fig. 13. Behavior of the Magnitude of the Gradient of the Training Error for Three Simulation Runs in Case 3 With α set to 0.90, 0.85, and 0.80, Respectively, Reflecting the Effects of Premature Saturation in the Output Unit for α values of 0.90 and 0.85

Fig. 14. Typical Activation Levels of the Output Unit of a Given Pattern in Case 3 for Three Simulation Runs When the Output Unit Prematurely Saturates to 1.0 for α values of 0.90 and 0.85

Fig. 15. Behavior of the Total Training Error in Case 3 for α=0.85 Obtained With the Proposed Approach and Slope Modification Approach to the Backpropagation Algorithm

Fig. 1a. At Iteration k-1 the Weight Update Δw_{k-1} Produces a Change in the Activation Level from o_{k-1} to o_k

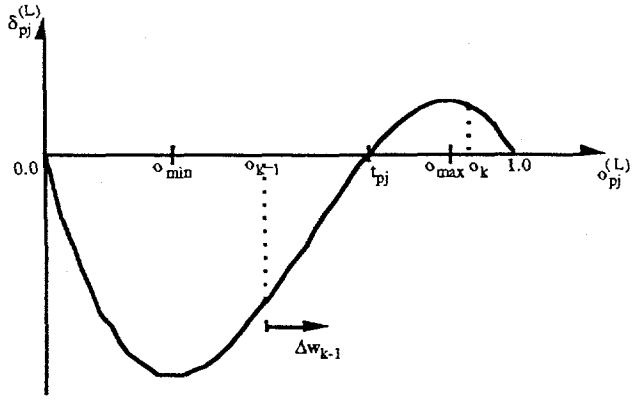

Fig. 1b. At Iteration k the Momentum Term $\alpha\Delta w_{k-1}$ Drives the Weight Update such that o_{k+1} is Further Away from the Target t_{pj}

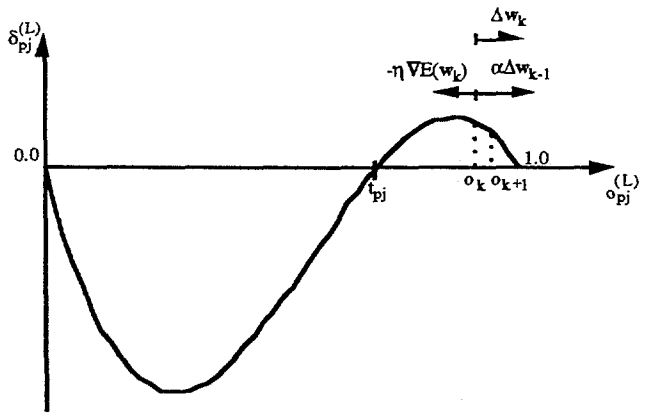

Fig. 1c. At Iteration k+1 the Momentum Term $\alpha\Delta w_k$ Drives the Weight Update such that o_{k+2} is Further Away from the Target t_{pj} and Closer to 1.0

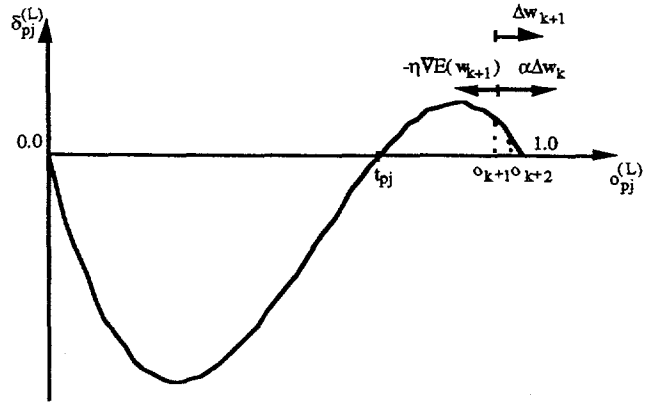

Fig. 2. Effects of the Premature Saturation of the Network Output Units in Case 1 on the Total Training Error

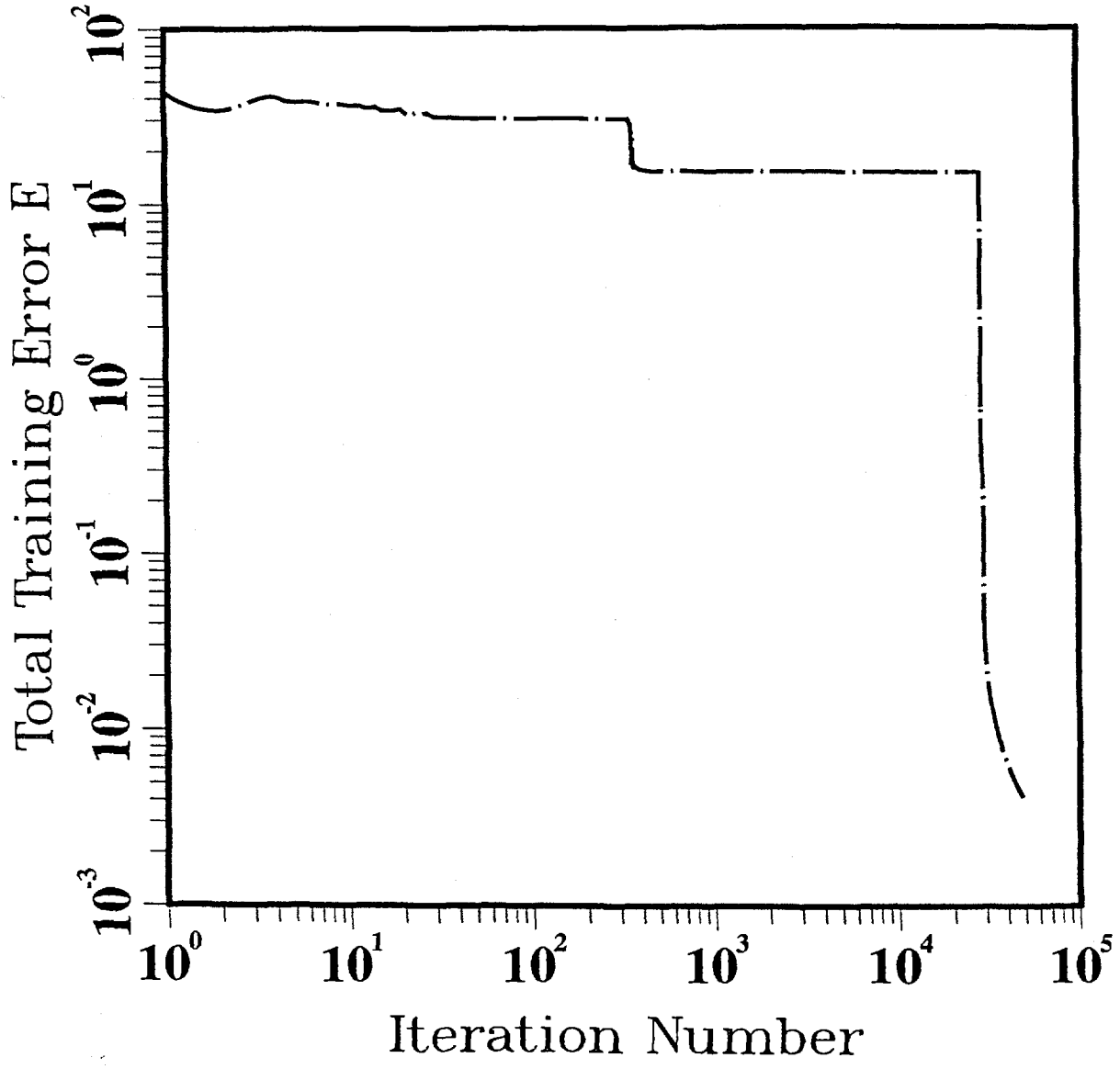

Fig. 3. Behavior of the Partial Training Errors for the Three Output Units in Case 1 Reflecting the Effects of Premature Saturation in Output Units 1 and 3

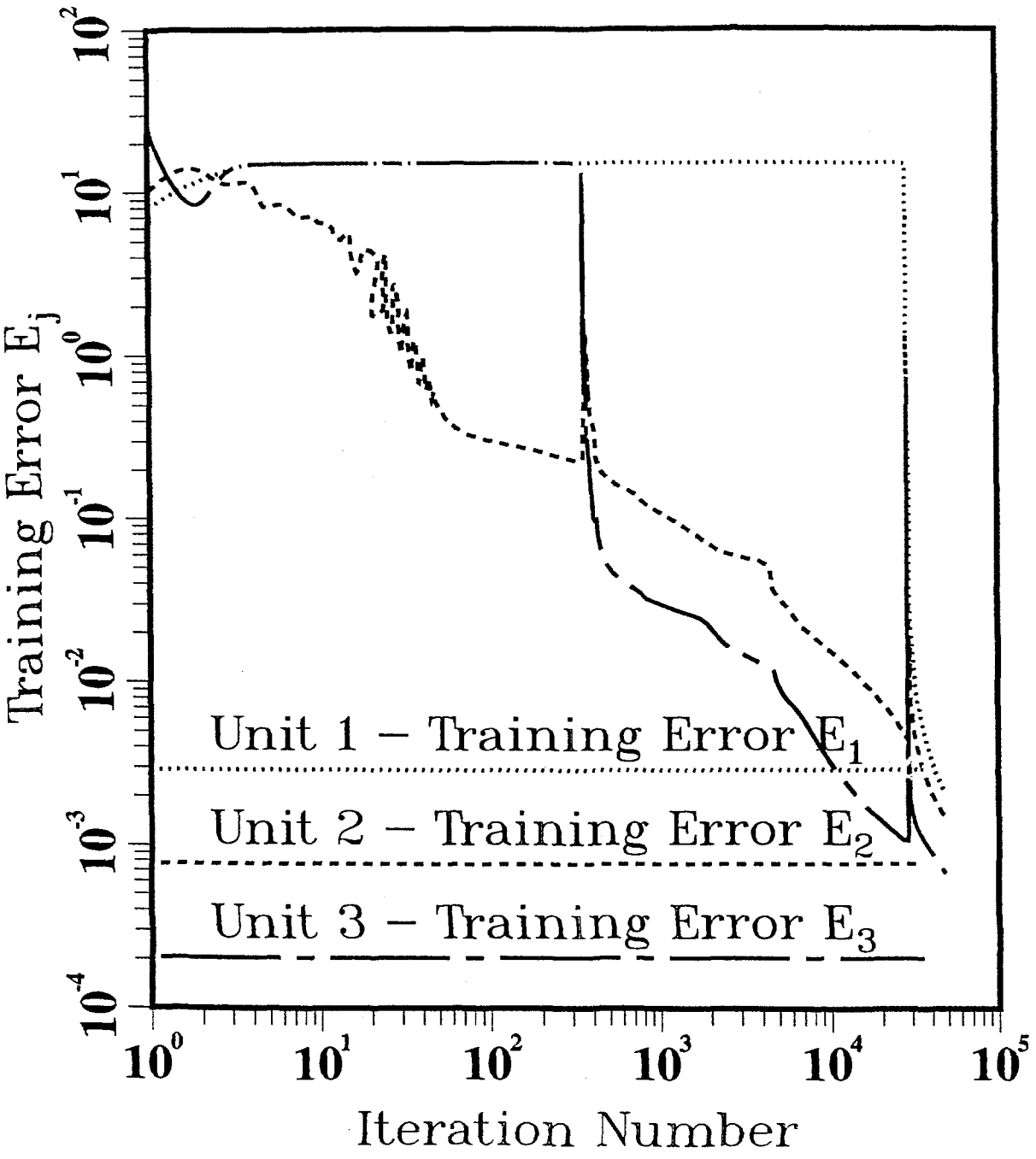

Fig. 4. Behavior of the Magnitude of the Gradient for the Partial Training Errors in Case 1 Reflecting the Effects of Premature Saturation in Output Units 1 and 3

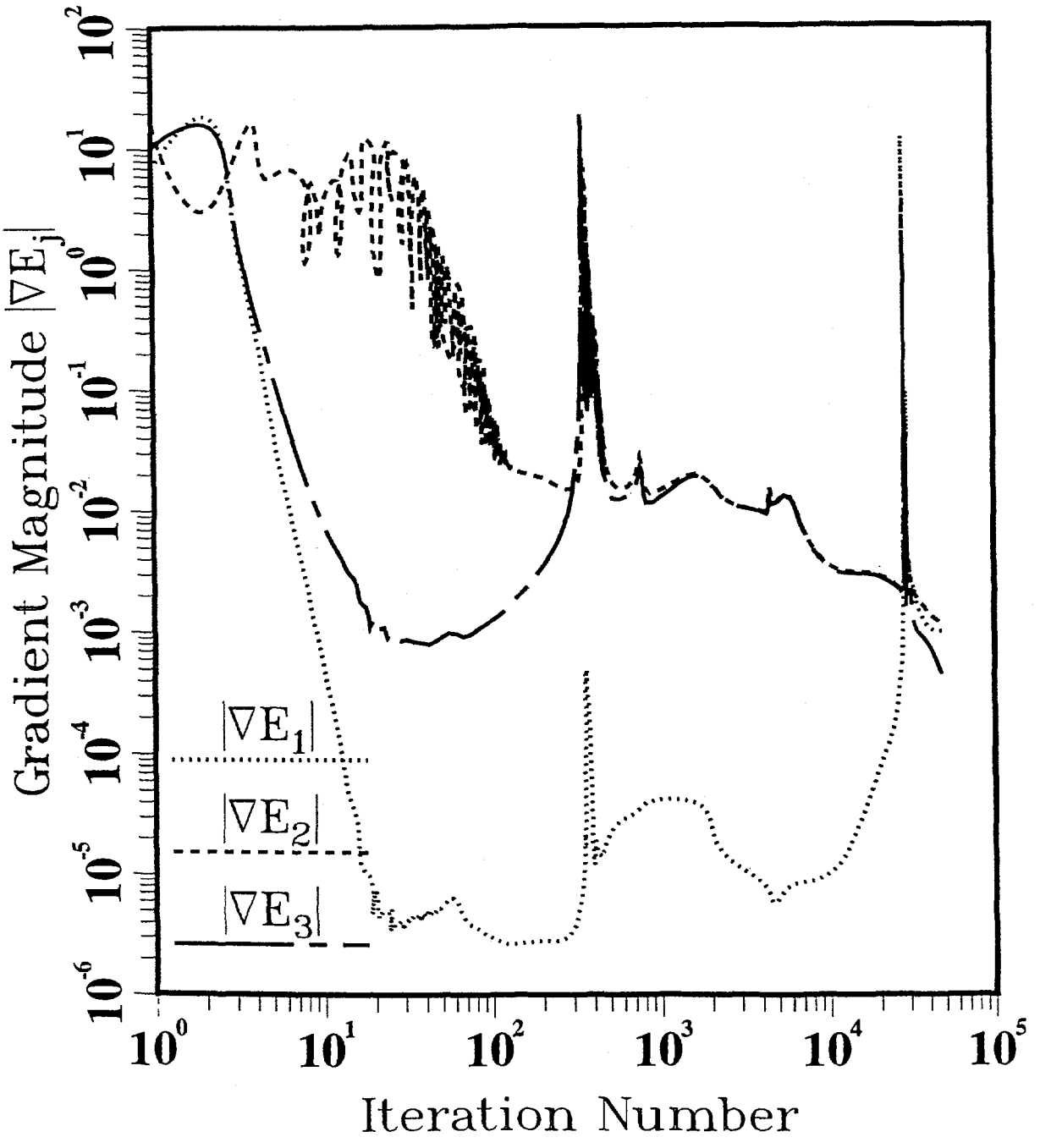

Fig. 5. Typical Activation Levels of the Three Output Units for a Given Pattern in Case 1 When Output Units 1 and 3 Prematurely Saturate to zero

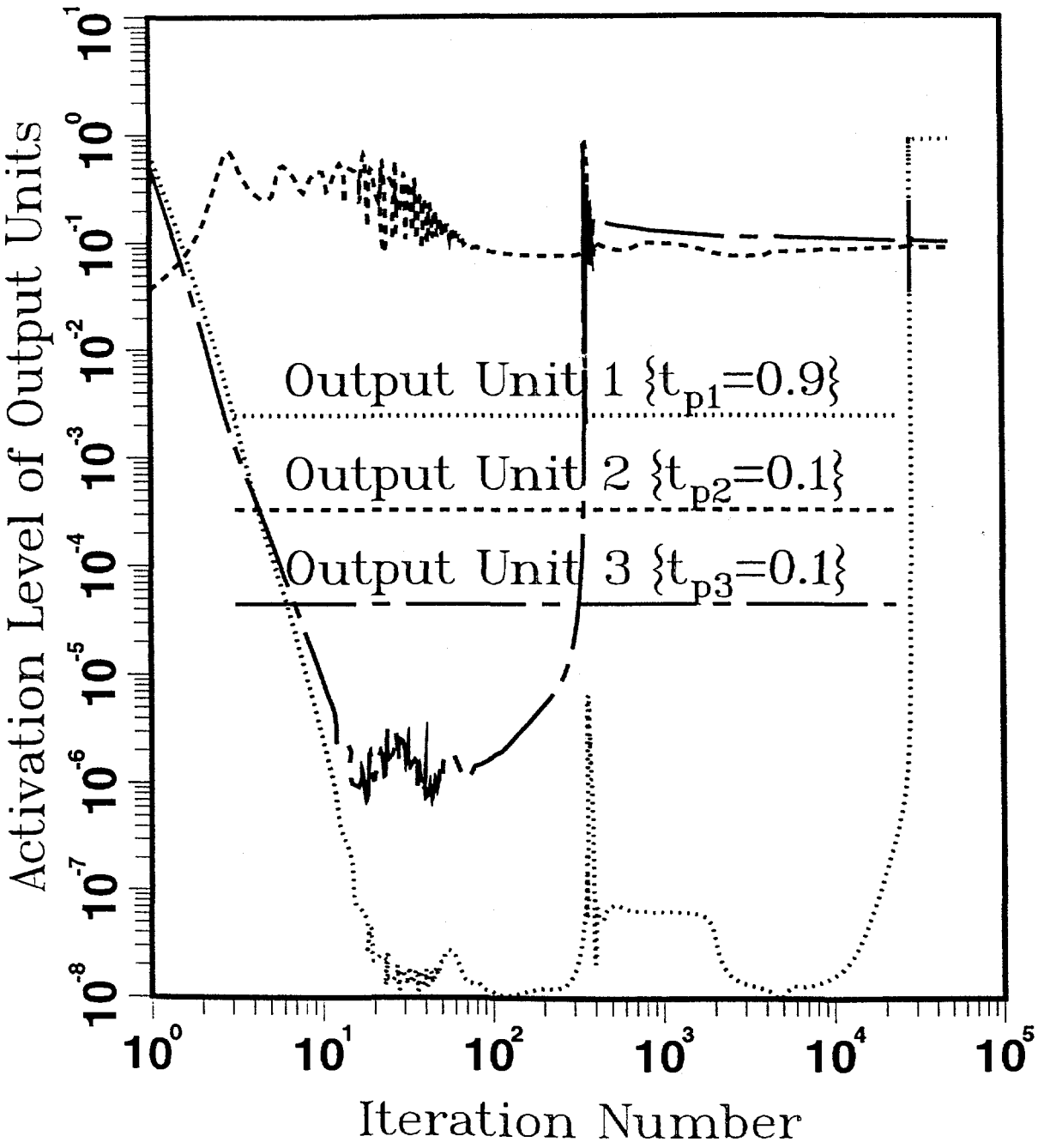

Fig. 6. Behavior of the Total and Partial Training Errors for the Three Output Units in Case 1 With the Proposed Modification to the Backpropagation Algorithm

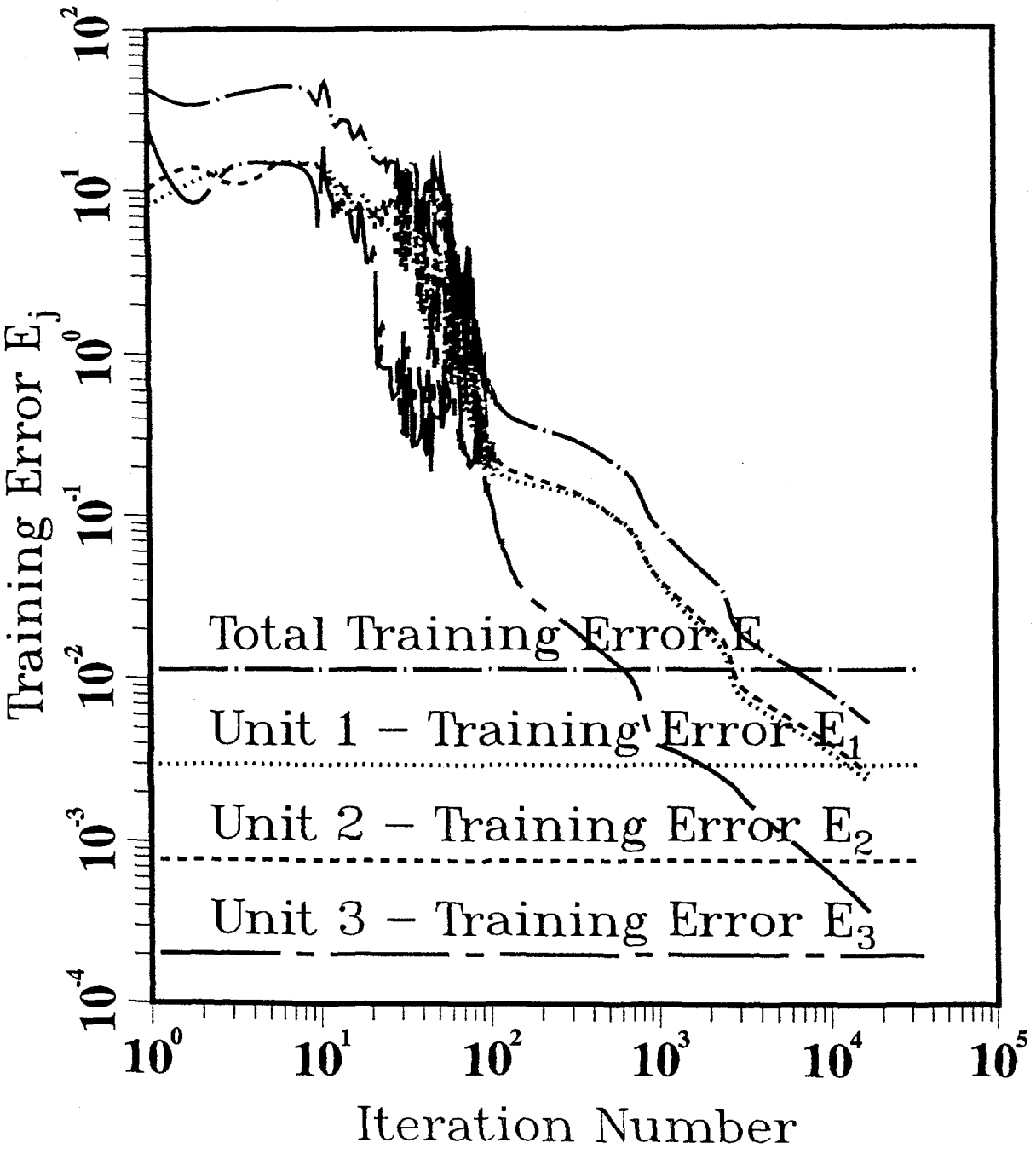

Fig. 7. Behavior of the Total and Partial Training Errors for the Three Output Units in Case 1 With the Slope Modification to the Backpropagation Algorithm

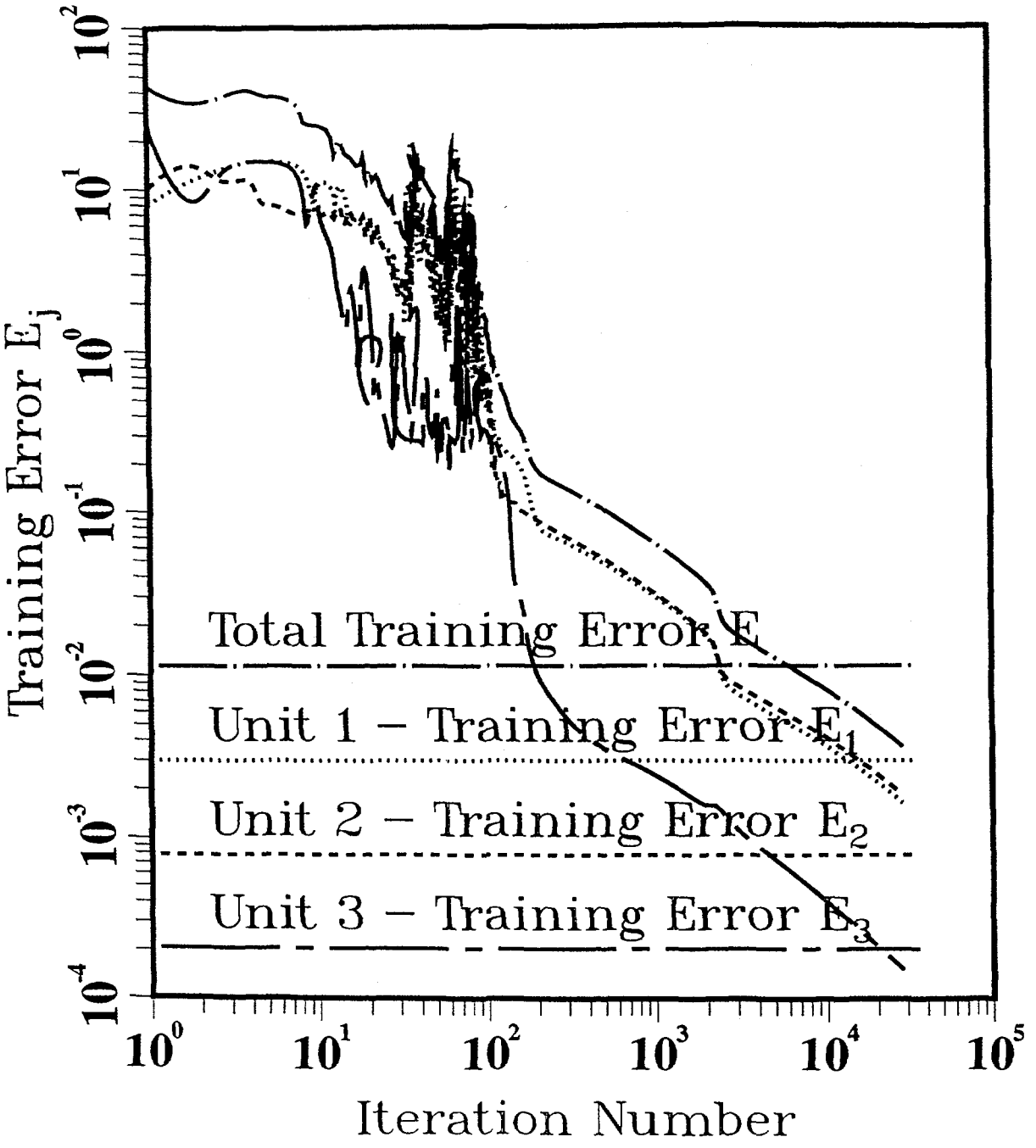

Fig. 8. Effects of the Premature Saturation of the Network Output Units in Case 2 on the Total Training Error

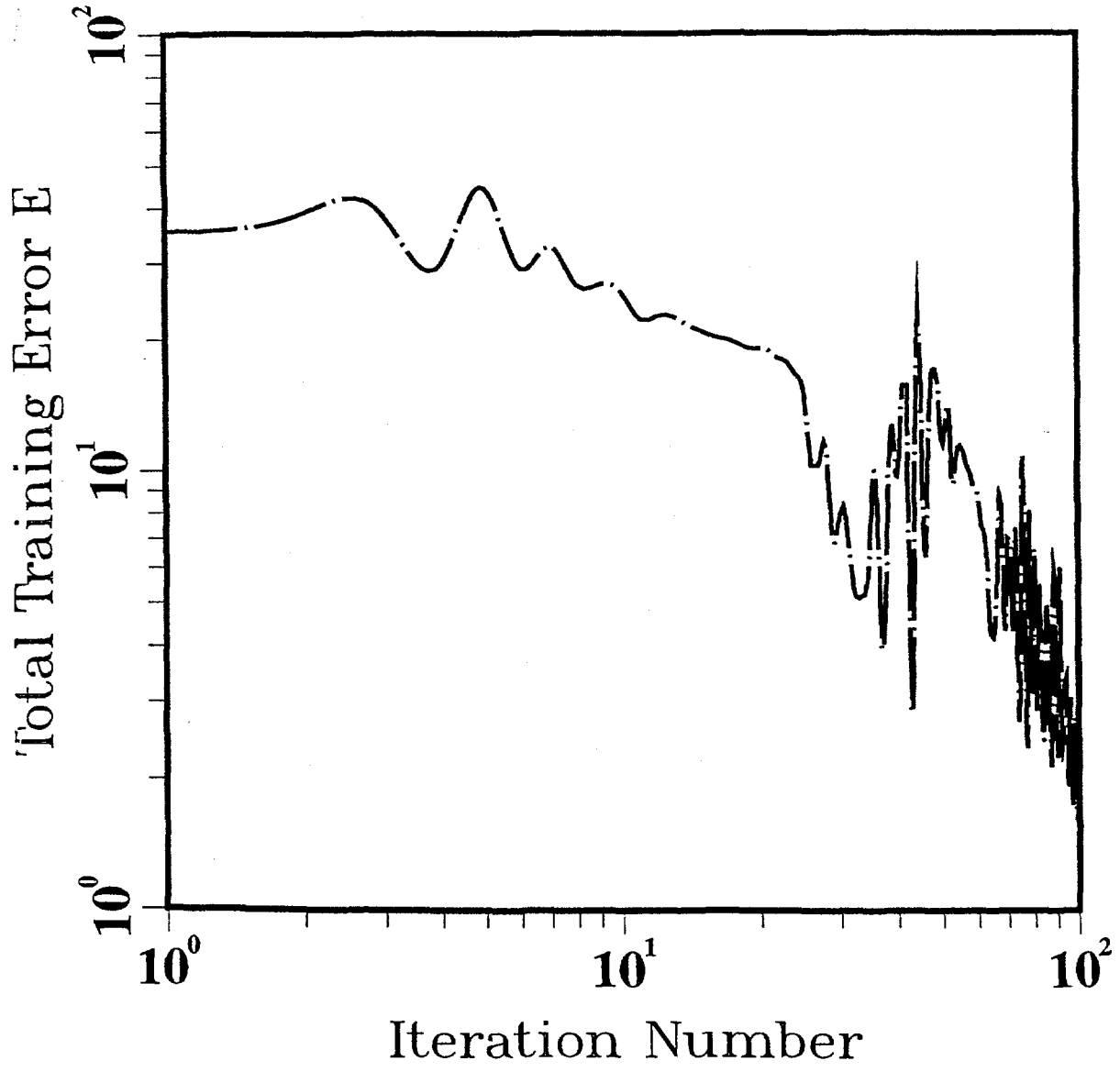

Fig. 9. Behavior of the Partial Training Errors for the Three Output Units in Case 2 Reflecting the Effects of Premature Saturation in Output Unit 2

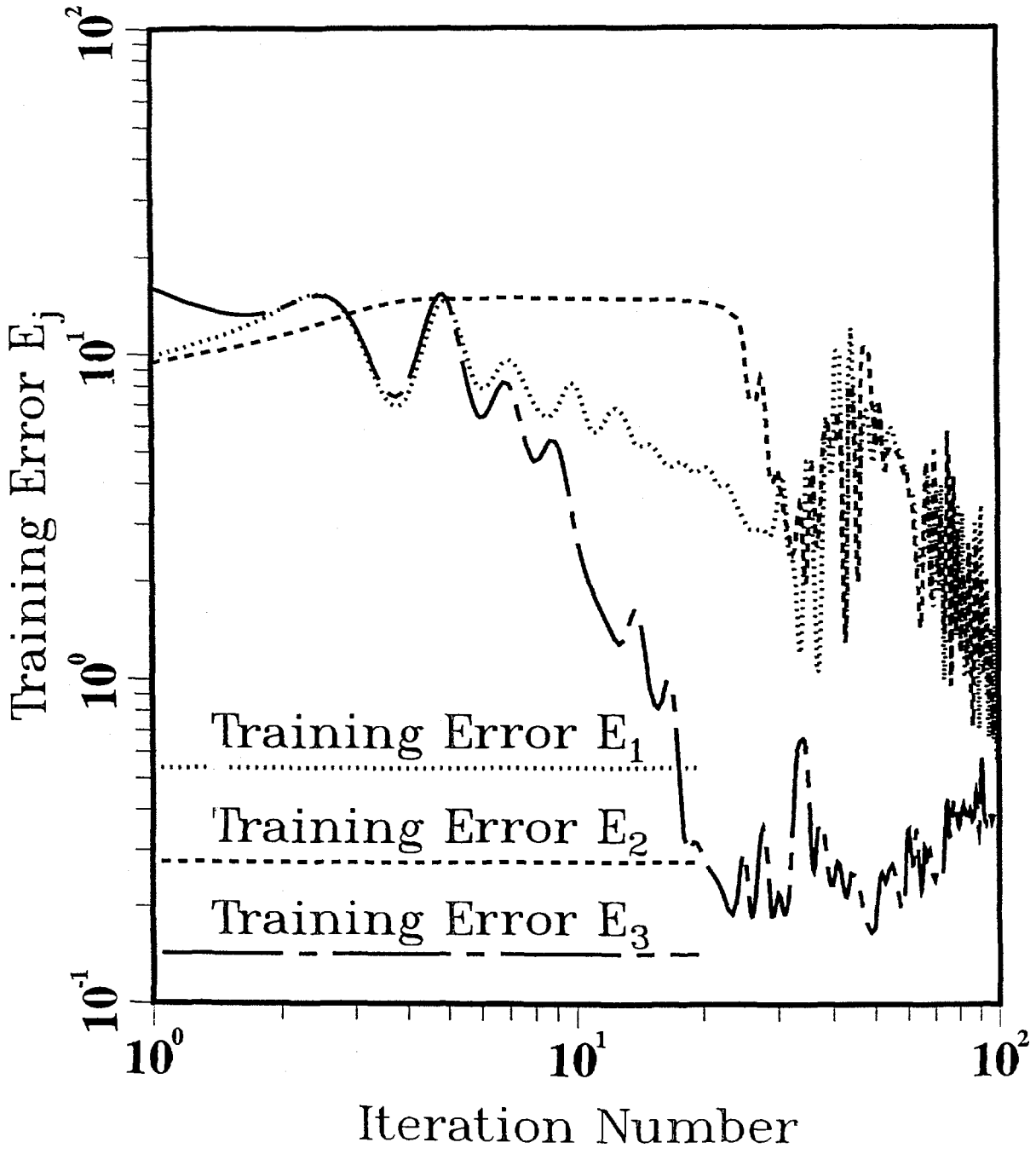

Fig. 10. Behavior of the Magnitude of the Gradient for the Partial Training Errors in Case 2 Reflecting the Effects of Premature Saturation in Output Unit 2

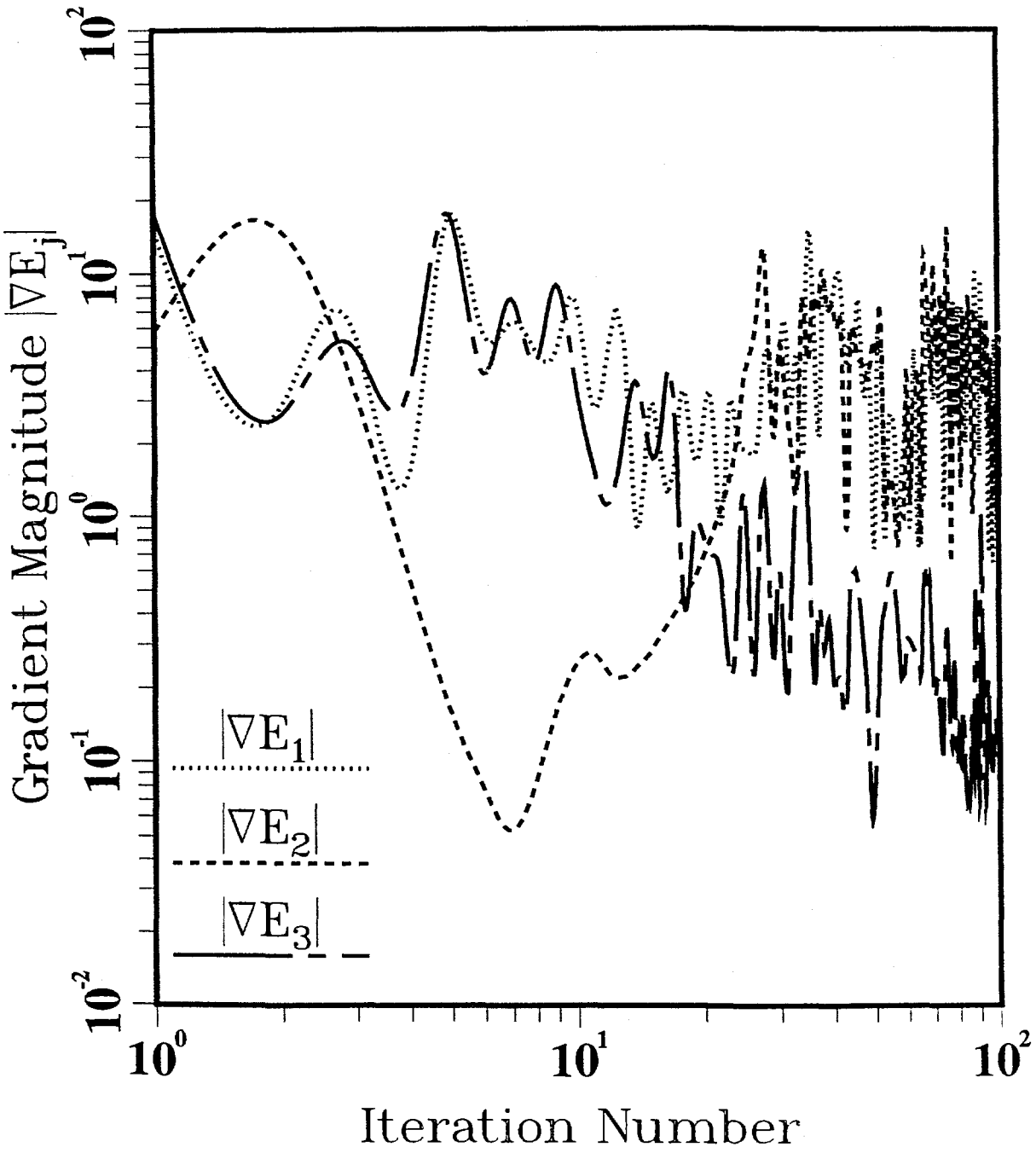

Fig. 11. Typical Activation Levels of the Three Output Units for a Given Pattern in Case 2 When Output Unit 2 Prematurely Saturates to zero

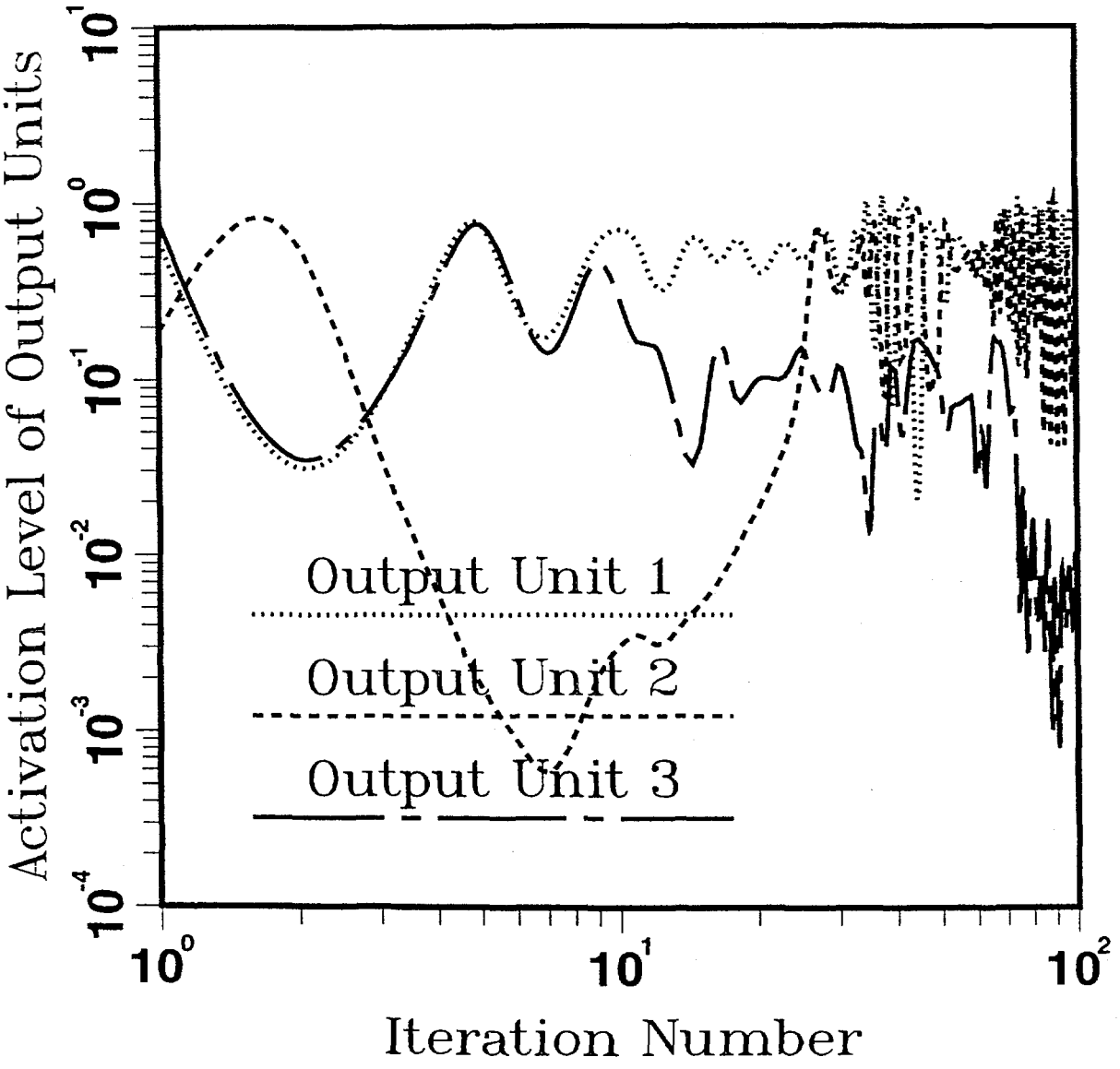

Fig. 12. Behavior of the Total Training Error for Three Simulation Runs in Case 3 With α set to 0.90, 0.85, and 0.80, Respectively, Reflecting the Effects of Premature Saturation for α values of 0.90 and 0.85

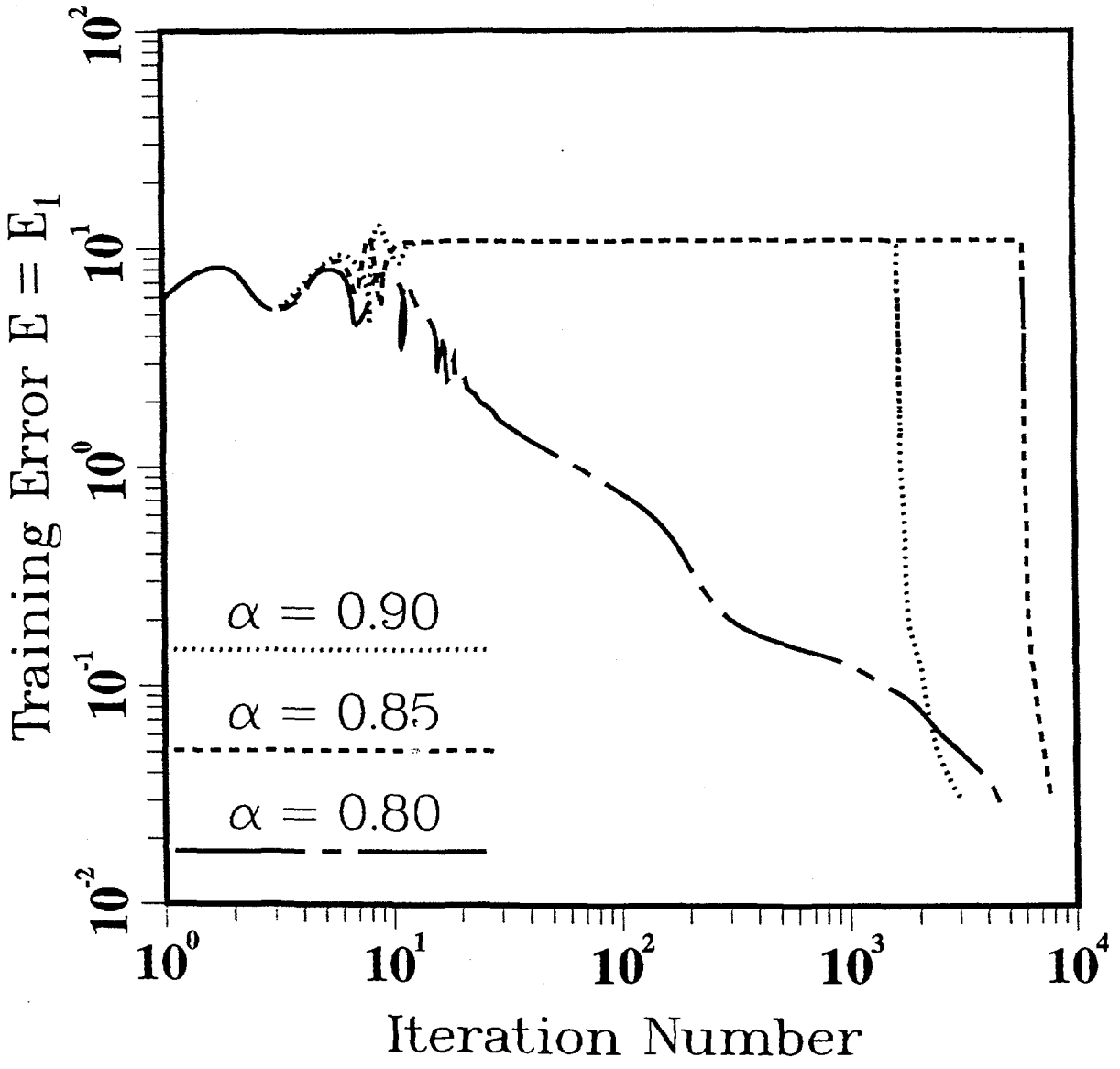

Fig. 13. Behavior of the Magnitude of the Gradient of the Training Error for Three Simulation Runs in Case 3 With α set to 0.90, 0.85, and 0.80, Respectively, Reflecting the Effects of Premature Saturation in the Output Unit for α values of 0.90 and 0.85

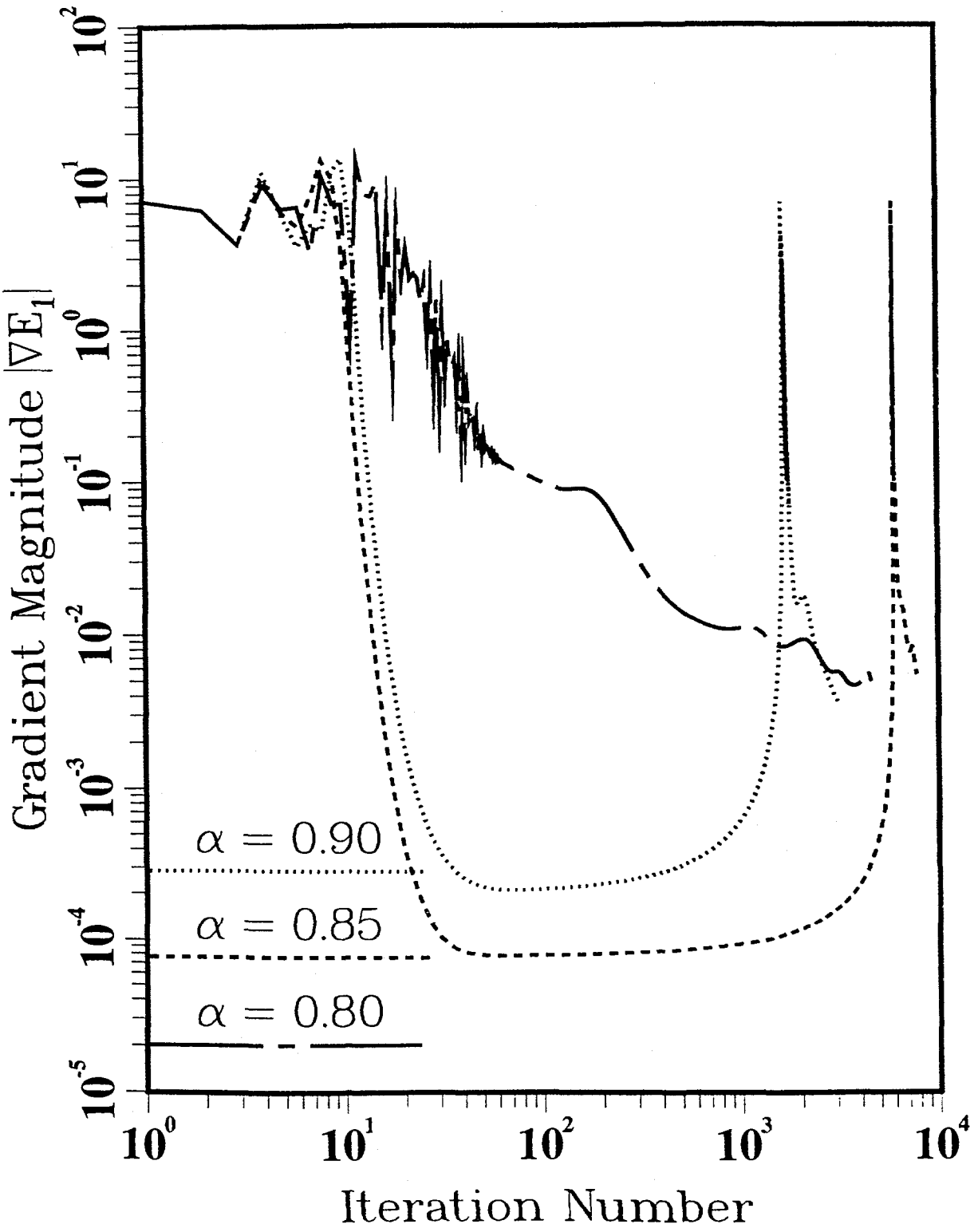

Fig. 14. Typical Activation Levels of the Output Unit of a Given Pattern in Case 3 for Three Simulation Runs When the Output Unit Prematurely Saturates to 1.0 for α values of 0.90 and 0.85

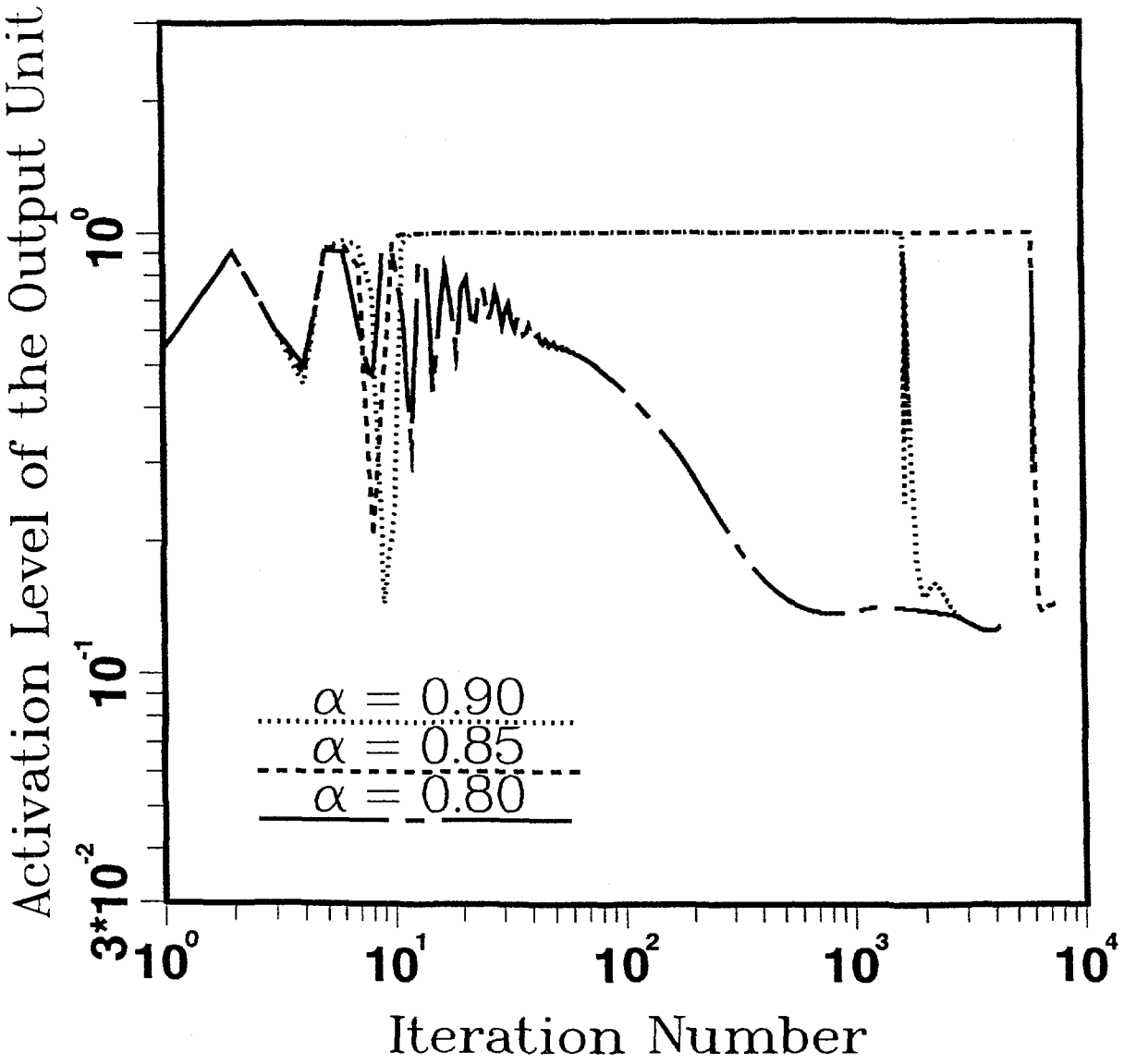

Fig. 15. Behavior of the Total Training Error in Case 3 for α=0.85 Obtained With the Proposed Approach and Slope Modification Approach to the Backpropagation Algorithm