Paper to be submitted to the Journal of Neural Computation.

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

Javier E. Vitela* and Jaques Reifman
Argonne National Laboratory
Reactor Analysis Division
Argonne, Illinois 60439

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 their validity is illustrated through simulation results.

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 that is sometimes overlooked, is the occurrence of the phenomenon of *premature saturation* (PS) of the network output units when the units are mapped by sigmoid functions (Franzini 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 result in tens to thousands of iterations which can strongly affect the already slow convergence of the BP algorithm.

* Currently on Sabbatical leave from: Instituto de Ciencias Nucleares, Universidad Nacional Autonoma de Mexico, 04510 Mexico D.F.

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 mechanism that gives rise to the occurrence of the phenomenon. The purpose of this work is to analyze this mechanism and to present the necessary conditions for its occurrence.

In the next Section a brief description of the standard BP training algorithm and the characterization of the phenomenon of premature saturation 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. Simulation results illustrating the validity of the necessary conditions are presented in Sec. 6, followed by a summary and conclusions in Sec. 7.

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. For reasons that will become clearer below, here we define E as the sum of partial training errors E_j ($j=1,2,...,J_L$) associated with each one of the J_L output units

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

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 ($p=1,2,...,P$). By this definition, we may recognize that everyone of the partial training errors E_j associated with the corresponding output unit j constitutes an individual error-surface $E_j(\mathbf{w})$ in weight-space.

At each iteration k, the weights **w** are updated through the weight-update rule (see, Rumelhart et al. 1986)

$$\Delta \mathbf{w}_k = -\eta \nabla E(\mathbf{w}_k) + \alpha \Delta \mathbf{w}_{k-1}, \tag{2}$$

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 $w_{ni}^{(\ell)}$ which connects the *i-th* unit in the *(ℓ-1)-th* layer with the *n-th* unit of the *ℓ-th* layer can be obtained recursively from the rule

$$\frac{\partial E_j}{\partial w_{ni}^{(\ell)}} = -\sum_{p=1}^{P} \delta_{pn}^{(\ell)} o_{pi}^{(\ell-1)}, \tag{3}$$

where $o_{pi}^{(\ell-1)}$ is the output or activation level of the *i-th* unit in the *(ℓ-1)-th* layer associated with the *p-th* teaching pattern and $\delta_{pn}^{(\ell)}$ is the delta rule associated with the training error E_j of the *j-th* output unit. For sigmoid mapping activation functions, the delta rule is given by:

$$\delta_{pn}^{(\ell)} = \begin{cases} [t_{pn} - o_{pn}^{(L)}]\, o_{pn}^{(L)}\, [1 - o_{pn}^{(L)}]; & \text{for } \ell = L \text{ and } n = j, \\ 0; & \text{for } \ell = L \text{ and } n \neq j, \\ o_{pn}^{(\ell)}\, [1 - o_{pn}^{(\ell)}] \sum_{m=1}^{J_{\ell+1}} \delta_{pm}^{(\ell+1)}\, w_{mn}^{(\ell+1)}; & \text{for } 1 < \ell < L. \end{cases} \tag{4}$$

This rule is similar to the original delta rule of Rumelhart et al. (1986), except that it 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 units of the last hidden layer and the output units. The affected weights $\mathbf{u}_j$ are those that are directly connected to output unit j.

In the literature, 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 all patterns 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. (4) corresponding to the slope of the sigmoid function approaches zero, causing the magnitude of the gradient components $\partial E_j / \partial w_{ni}^{(\ell)}$ to attain small values. The weights $\mathbf{u}_j$ 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. The 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 high error levels.

In the past, these plateaus have been erroneously interpreted by some authors (see, e.g., Dahl 1987) as an intrinsic process used by neural networks while constructing internal representations to distinguish between different input patterns. Further research has shown that the only role of PS is to produce a detrimental effect in the training process, which is manifested as an increment in the number of training cycles required to release the trapped weights from their saturated state.

A number of researchers have addressed the problems posed by the premature saturation 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. (4) 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 work best is the one where 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. (4), 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. In their analysis, the PS of the output units of the network is described as a static phenomenon that occurs at the first training cycle as a consequence of the randomly chosen initial set of weights. They also presented 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 the dynamic mechanism that produces the phenomenon of PS, followed by a set of necessary conditions that need to be satisfied for the occurrence of the phenomenon.

3 The Mechanism that Produces Premature Saturation

Premature saturation of a given output unit j is likely to occur at the early stages of the training process when the randomly selected weights, i.e., the starting point of the BP algorithm, lies in a skewed region of the error-surface E_j. The skewness of this error-surface may cause the components of the gradient $\nabla E(\mathbf{w}_k)$ associated with the subset of weights $\mathbf{u}_j$ defined in the last Section, to change signs at two consecutive iterations of the algorithm. In that case, at iteration k, the momentum term $\alpha\Delta\mathbf{w}_{k-1}$ in Eq. (2) which represents the "memory" of the previous directions of motion, may not have its projection along the direction of the gradient ∇E_j, pointing in the desired negative gradient direction $-\nabla E_j$. If in addition, this projection is larger in magnitude than the corresponding projection of the learning term $-\eta\nabla E(\mathbf{w}_k)$, then the new weight update $\Delta\mathbf{w}_k$ will produce, at first order, an increase in the training error $E_j(\mathbf{w}_{k+1})$.

The observation that PS generally occurs at the early stages of the training process is due to the fact that at early stages the activation level of the output units are far from their target values causing the components of the weight update $\Delta\mathbf{w}_k$ to attain relatively large absolute values. Similarly, PS is more often observed with the "batch" mode of BP

training (Becker and Le Cun 1988), rather than the "on-line" mode because of the larger Δw_k obtained by adding up the contributions of all patterns. When the components of Δw_k are of the same order of magnitude as the components of w_k, the updated weight $w_{k+1}=w_k+\Delta w_k$ may suddenly become quite large and cause the activation level $o_{pj}^{(L)}$ of the *j-th* output unit to approach either 0 or 1, for all patterns p=1,2,...,P, in the next pattern presentation, i.e., iteration k+1, of the BP algorithm. The combination of the effects of this scenario with the effects of a skewed error-surface E_j may cause an increase in $E_j(w_{k+1})$ and a decrease in the magnitude of $\nabla E_j(w_{k+1})$, reducing the contribution of the latter to the total gradient $\nabla E(w_{k+1})$. If in addition, the new learning term $-\eta\nabla E(w_{k+1})$ cannot offset the tendency of the new momentum term $\alpha\Delta w_k$ to update the weights in the direction in which E_j increases, the magnitude of $\nabla E_j(w_{k+2})$ will be further reduced in the subsequent iteration.

As the magnitude of ∇E_j is further reduced, so is its contribution to ∇E allowing the momentum term to keep governing the motion of the weights across the error-surface E_j. The repetition of this undesirable mechanism for consecutive iterations launches a "snowball" effect that causes, as an end-result, the components of ∇E_j to approach an asymptotic zero value. As a consequence, the weights $\mathbf{u}_j$ connecting all units of the last hidden layer to output unit j become trapped at their current values and the activation level $o_{pj}^{(L)}$ remains at its saturated state. This precludes any significant change in E_j originating the characteristic flat plateau in the training error curve.

As ∇E_j decreases quickly towards its asymptotic zero value during the snowball effect, so does the contribution of the learning term $-\eta\nabla E$ in updating $\mathbf{u}_j$. Because of this and the fact that the momentum parameter α is selected from the [0,1] interval, the contribution of the momentum term $\alpha\Delta w$ in updating $\mathbf{u}_j$ also decreases, but at a slower rate. The contribution of $\alpha\Delta w$ decreases continuously at each iteration beyond the end of the snowball effect until it becomes comparable with the asymptotic contribution of $-\eta\nabla E$, at which point the trapped weights $\mathbf{u}_j$ may start recovering from their frozen state.

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, and the number and type of training patterns. These dependencies together with the strong nonlinearity of sigmoid-mapped units offer tremendous difficulties in obtaining both the necessary and sufficient conditions for the occurrence of premature saturation. Hence, in the following Section we present only a set of *necessary* conditions that need to be satisfied if an output unit is to saturate prematurely.

4 Necessary Conditions for Premature Saturation

Based on the above discussions, we established the following four necessary conditions which must be satisfied if premature saturation is to occur in the *j-th* output unit:

(c1) $\eta \nabla E(w_k) \cdot \nabla E_j(w_k) - \alpha \Delta w_{k-1} \cdot \nabla E_j(w_k) < 0,$

(c2) $\Delta w_{k-1} \cdot \nabla E_j(w_k) > 0,$

(c3) $\alpha \, |\Delta w_{k-1} \cdot \nabla E_j(w_k)| >> \eta \, |\nabla E(w_k) \cdot \nabla E_j(w_k)|,$ and

(c4) $|\nabla E_j(w_{k+1})| < |\nabla E_j(w_k)|.$

The mechanism that gives rise to premature saturation is embedded in the four necessary conditions. To first order, each iteration of the BP algorithm yields smaller values of E_j if the weight update Δw_k satisfies the inequality $\Delta w_k \cdot \nabla E_j(w_k) < 0$. However, as discussed in the previous Section, at the onset of PS this inequality is not satisfied, i.e., E_j increaes, which is expressed by condition c1. Condition c2 states that the projection of the momentum term along the direction $\nabla E_j(w_k)$ should be in the direction which causes E_j to increase. Condition c3 reflects the necessity that the magnitude of the projection of the momentum term must be larger than the projection of the learning term, along the ∇E_j direction. These last two conditions summarize the role of the momentum term $\alpha \Delta w_{k-1}$ as the driving force that governs the updating of the weights at the onset of premature saturation. Finally, condition c4 requires that the magnitude of the gradient of unit j decreases monotonically at subsequent iterations and reflects the fact that the activation level of the unit is prematurely approaching the saturated levels of 0 or 1.

As mentioned before, these conditions, although necessary, are not sufficient. Nevertheless, our experimental results have shown that the satisfaction of these four conditions for a large enough number of successive iterations characterizes the phenomenon unequivocally. The number of successive iterations for which the conditions need to be satisfied in order to saturate output unit j is problem-dependent and the larger the number of successive iterations is, the higher is the degree of saturation that the unit may reach. The degree of saturation that a saturating unit j reaches can be quantified by the magnitude of the gradient ∇E_j at the last iteration in which the four conditions are satisfied. Thus, smaller asymptotic zero values of $|\nabla E_j|$ indicate higher degrees of saturation and vice-versa.

5 Stages of the Process of Premature Saturation

The analysis presented in the foregoing Sections allow us to characterize the entire process of premature saturation of output unit j by three distinct stages: beginning of saturation, saturation plateau, and recovery from saturation. The first stage, beginning of saturation, corresponds to the first few iterations of the BP algorithm where conditions c1 through c4 are simultaneously satisfied during consecutive iterations. During this stage, the "snowball" effect is launched causing $|\nabla E_j|$ to decrease monotonically and to approach an asymptotic value of zero.

The second stage, saturation plateau, starts when the four conditions are no longer simultaneously satisfied and both the momentum term and learning term have become small enough to produce any significant update of the weights $\mathbf{u}_j$ at a single iteration. The trapping of the weights causes the training error E_j to remain practically constant, which is then reflected as a saturation plateau in the training curve at high values of the total training error E. The length of the saturation plateau is strongly dependent on the level of saturation of the unit, with long lengths associated with high levels of saturation and vice-versa. If the level of saturation characterized by $|\nabla E_j|$, is not high enough the unit may only show a slight tendency to saturate in which case the saturation plateau may not be noticeable. Also, during the second stage, $|\nabla E_j|$ suffers fluctuations about its asymptotic zero value and in some cases during this stage the four necessary conditions may once again become satisfied for a number of iterations.

Finally, the third and last stage, recovery from saturation, begins when the magnitude of the gradient $|\nabla E_j|$ starts increasing monotonically. The monotonic increase is a consequence of the fact that the learning term and the momentum term are now allowed to update the weights in the direction in which E_j decreases; thus untrapping the weights $\mathbf{u}_j$ and reversing the snowball effect described in Sec. 3 during the saturation of the output unit. The end of the recovery from saturation stage is characterized by values of $|\nabla E_j|$ of the same order of magnitude of its values just before the beginning of the saturation stage followed by a quick settle down.

6 Simulation Results

In order to show the validity of the necessary conditions described in Sec. 4, we present the results of a BP training session for the classification of three component failures in a nuclear power plant in which the phenomenon of premature saturation was observed in the network output units (Reifman and Vitela 1994). The network consisted of three layers with 20-20-3 units per layer, respectively, and 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 0.0 for the first two training cycles and after that it was set to 0.9. A training cycle or 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 adjusted.

Figure 1 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 network three output units, to saturate prematurely. The occurrence of the premature saturation 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 of the network is illustrated in Fig. 2. Figure 2 shows that output units 1 and 3 are the two saturated units responsible for the formation of the flat plateaus in Fig. 1 and that the sharp decrease in the total training error E around 350 training cycles 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 satisfied. As illustrated in Fig. 2, 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 satisfy necessary condition c1 and expresses the fact that the weights directly connected to units 1 and 3 are not being updated in a minimizing direction, as a consequence of the satisfaction of necessary conditions c2 and c3. Necessary condition c4, which requires that the magnitude of the gradient $|\nabla E_j|$ for saturating unit j decreases monotonically at subsequent iterations, is also satisfied after the second iteration for both output units as illustrated in Fig. 3.

Table I summarizes the results of testing the three output units for the four necessary conditions during the first 20 iteration of the BP algorithm. Each entry in the table is a number with four binary digits, where each one of the four digits corresponds to the status of one necessary condition. A necessary condition is satisfied if the associated digit has a value of 1, and is not satisfied if the value of the digit is 0. For example, the entry for unit 2 at the fourth iteration, 0101, satisfies conditions c2 and c4 but does not satisfy conditions c1 and c3. The table shows that in both units 1 and 3 the four necessary conditions are satisfied simultaneously between iterations 3 and 14 constituting the beginning of saturation stage for both output units.

The second stage of the saturation process, in which E_1 and E_3 remain practically constant, starts at iteration 15 for both saturated units. At this point, at least one of the four conditions is no longer satisfied, although the magnitude of the gradients $|\nabla E_1|$ and $|\nabla E_3|$, which quantify the level of saturation of the units, are still decreasing and approaching their

asymptotic "zero" value. The duration of this stage is different for each unit due to the different level of saturation reached by each output unit at the end of the first stage. The fact that output unit 1 has a higher degree of saturation than output unit 3, as illustrated in Fig. 3, causes unit 1 to delay its recovery from its PS state. This is manifested by a larger saturation plateau in Fig. 2.

The recovery from saturation stage beings around iteration 70 for output unit 3, while for unit 1 it only starts around iteration 6000. This final stage of the saturation process is characterized by the monotonic increase in the magnitude of ∇E_j, which is satisfied until the recovery from saturation is completed. As shown in Fig. 3, the full recovery from saturation occurs around iteration 350 for unit 3 and around iteration 30000 for output unit 1, after which $|\nabla E_j|$, for j=1,3, decreases as the BP algorithm is free from the phenomenon of PS and resumes its motion towards the minimum of the total training error. The typical activation levels of the three output units corresponding to an arbitrary teaching pattern are illustrated in Fig. 4.

The results presented for this particular training session were confirmed in other experiments performed by changing the learning parameter, momentum parameter, and by starting the training session at different positions in weight-space. These experiments have also shown that the occurrence of PS in a given output unit is an overall property of the network, with the remaining output units playing important roles. Because all output units contribute to the total error gradient $|\nabla E|$ and therefore to the weight update Δw_k, the unsaturated output units may alter the activation level of the units in the last hidden layer, which in turn, may alter the activation level of the saturated output units. In this way, the unsaturated units may prevent a saturated unit from reaching high degrees of saturation during the early stages of the saturation process, while on the other hand, they may inhibit a faster recovery of the saturated unit at later stages.

7 Summary and Conclusions

In this work we have analyzed the mechanism by which the process of premature saturation 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 was established and it was concluded that the momentum term plays the leading role at the onset of premature saturation. Furthermore, our analysis showed that the entire premature saturation process can be characterized by three distinct stages, beginning of saturation, saturation plateau, and the recovery from saturation stage.

The validity of these results was illustrated by means of a typical premature saturation problem encountered during a training session of a feedforward multilayer

network. Additional experiments performed changing the values of the learning parameter, momentum parameter, as well as running the same problem with different sets of initial weights confirmed the results presented here and validate the necessary conditions. Finally, we want to point out that although not sufficient, the satisfaction of these conditions for a large enough number of consecutive iterations constitutes an unequivocal sign of premature saturation.

Acknowledgments

The second author was supported by the U.S. Department of Energy, Nuclear Energy Program, under contract number W-31-109-ENG-38.

References

Balakrishnan, K., and Honavar, V., "Improving Convergence of Back-Propagation by Handling Flat-Spots in the Output Layer," *Proc. of the International Conference on Artificial Neural Networks*, Brighton, United Kingdom, September 4-7, 1992, p. 1003, vol. 2, I. Aleksander and J. Taylor, Eds., Elsevier Science (1992).

Becker, S., and Le Cun, Y., "Improving the Convergence of Back-Propagation Learning with Second Order Methods," *Proc. 1988 Connectionist Models Summer School*, Carnegie-Mellon University, p. 29, D. Touretzky, G. Hinton, and T. Sejnowski, Eds., M. Kaufmann, San Mateo, California (1988).

Chen, J. R., and Mars, P., "Stepsize Variation Methods for Accelerating the Back-Propagation Algorithm," *Proc. International Joint Conference on Neural Networks*, Washington D. C., January 15-19, 1990, p. 601, vol. I, M. Caudill, Ed., IEEE Neural Networks Council, Piscataway, NJ (1990).

Dahl, E. D. "Accelerated Learning Using the Generalized Delta Rule," *Proc. IJCNN First Int. Conf. Neural Networks*, San Diego, California, June 21-24, 1987, Vol. II, P. 523, M. Caudill and C. B. Butler, Eds., SOS Print, Piscataway, NJ (1987).

Fahlman, S. E., "Faster-Learning Variations on Back-Propagation: An Empirical Study," *Proc. of the 1988 Connectionist Models Summer School*, Carnegie-Mellon University, p. 38, D. Touretzky, G. Hinton, and T. Sejnowski, Eds., M. Kaufmann, San Mateo, CA (1989).

Franzini, M. A., "Speech Recognition With Back Propagation," *Proc. of the Ninth Annual Conference of the IEEE Engineering in Medicine and Biology Society*, p. 1702, vol. 3, 1987.

Lee, Y., Oh, S. H., and Kim, M. W., "The Effect of Initial Weights on Premature Saturation in Back-Propagation Training," *Proc. Int. Joint Conf. Neural Networks*, Seattle, Washington, July 8-12, 1991, Vol. I, p.I-765, Staff Eds., IEEE Neural Networks Council (1991).

Parekh, R., Balakrishnan, K., and Honavar,V., "An Empirical Comparison of Flat-Spot Elimination Techniques in Back-Propagation Networks," *Proc. of the Third Workshop on Neural Networks: Academic/Industrial/NASA/Defense*, Auburn, Alabama, February 10-12, 1992, p. 55, Society for Computer Simulation, San Diego, CA (1993).

Reifman, J., and Vitela, J. E., "Accelerating Learning of Neural Networks with Conjugate Gradients for Nuclear Power Plant Applications," *Nucl. Technol.*, **106**, 225 (1994).

Rumelhart, D. E., Hinton, G. E. and Williams, R. J., "Learning Internal Representations by Error Propagation," Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. I, p. 319, D. E. Rumelhart and J. C. McClelland, Eds., The MIT Press, Cambridge, Massachusetts (1986).

Spartz, R., and Honavar, V., "An Empirical Analysis of the Expected Source Values Rule," *Proc. of the Third Workshop on Neural Networks: Academic/Industrial/NASA/ Defense*, Auburn, Alabama, February 10-12, 1992, p. 95, Society for Computer Simulation, San Diego, CA (1993).

Vitela, J., and Reifman, J., "Enhanced Backpropagation Training Algorithm for Transient Event Identification," *Trans. Am. Nucl. Soc.*, **69**, 148 (1993).

Table I. Results of the Four Necessary Conditions for the Three Output Units During the First Few Training Cycles

Training Cycle	Output Unit 1	Output Unit 2	Output Unit 3
1	0000	0001	0000
2	0001	0000	0001
3	1111	0000	1111
4	1111	0101	1111
5	1111	1110	1111
6	1111	0101	1111
7	1111	0101	1111
8	1111	0010	1111
9	1111	0101	1111
10	1111	0100	1111
11	1111	0100	1111
12	1111	0101	1111
13	1111	0010	1111
14	1111	0100	1111
15	1101	0101	1101
16	1111	0101	1111
17	1111	0000	1111
18	1101	0100	1101
19	0100	0101	0101
20	1001	0101	1001

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

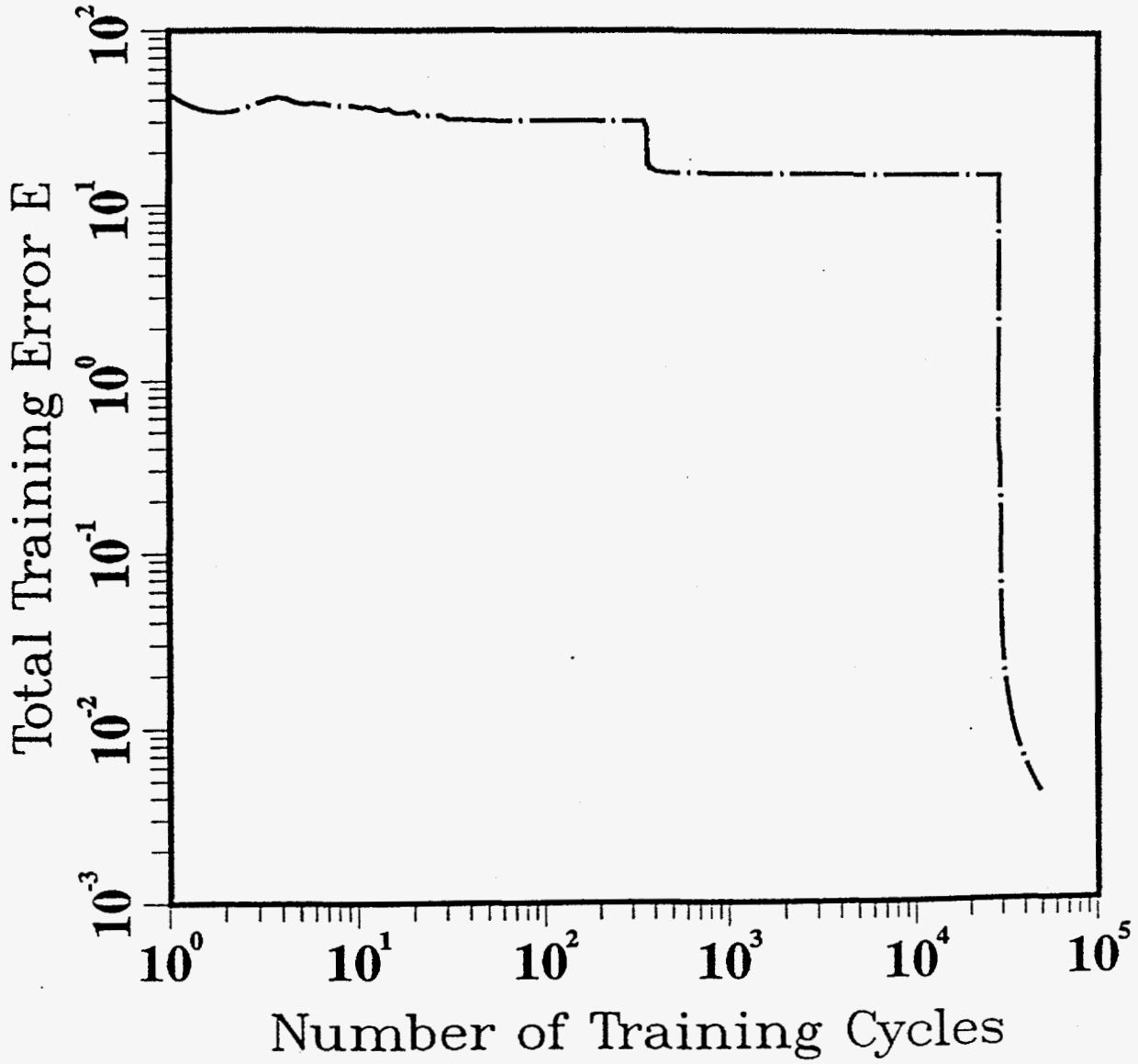

Fig. 2. Behavior of the Partial Training Errors Associated With the Three Output Units Showing the Effect of Premature Saturation in Two Units

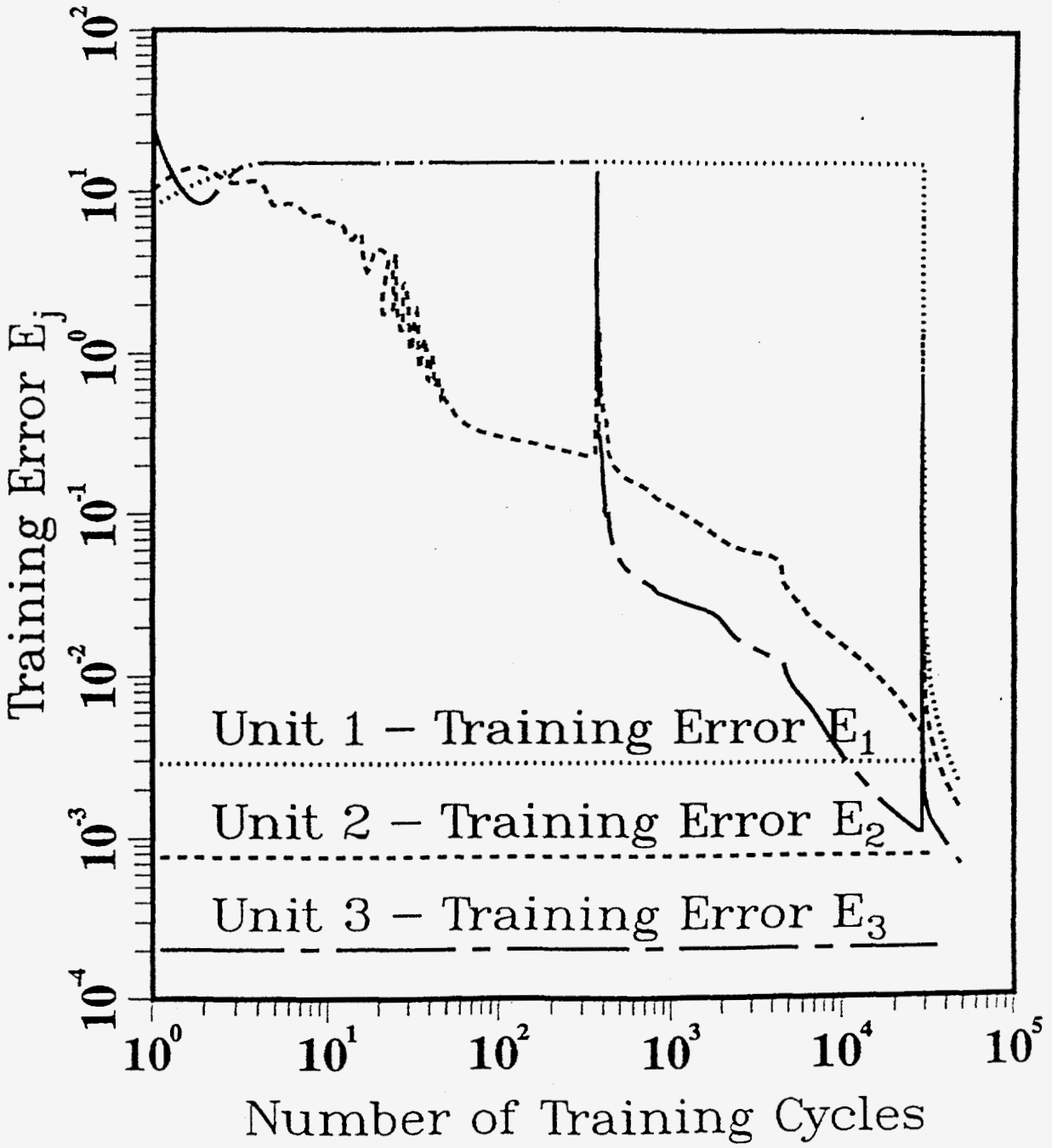

Fig. 3. Behavior of the Magnitude of the Gradient of the Partial Training Errors Showing the Effect of Premature Saturation in Output Units 1 and 3

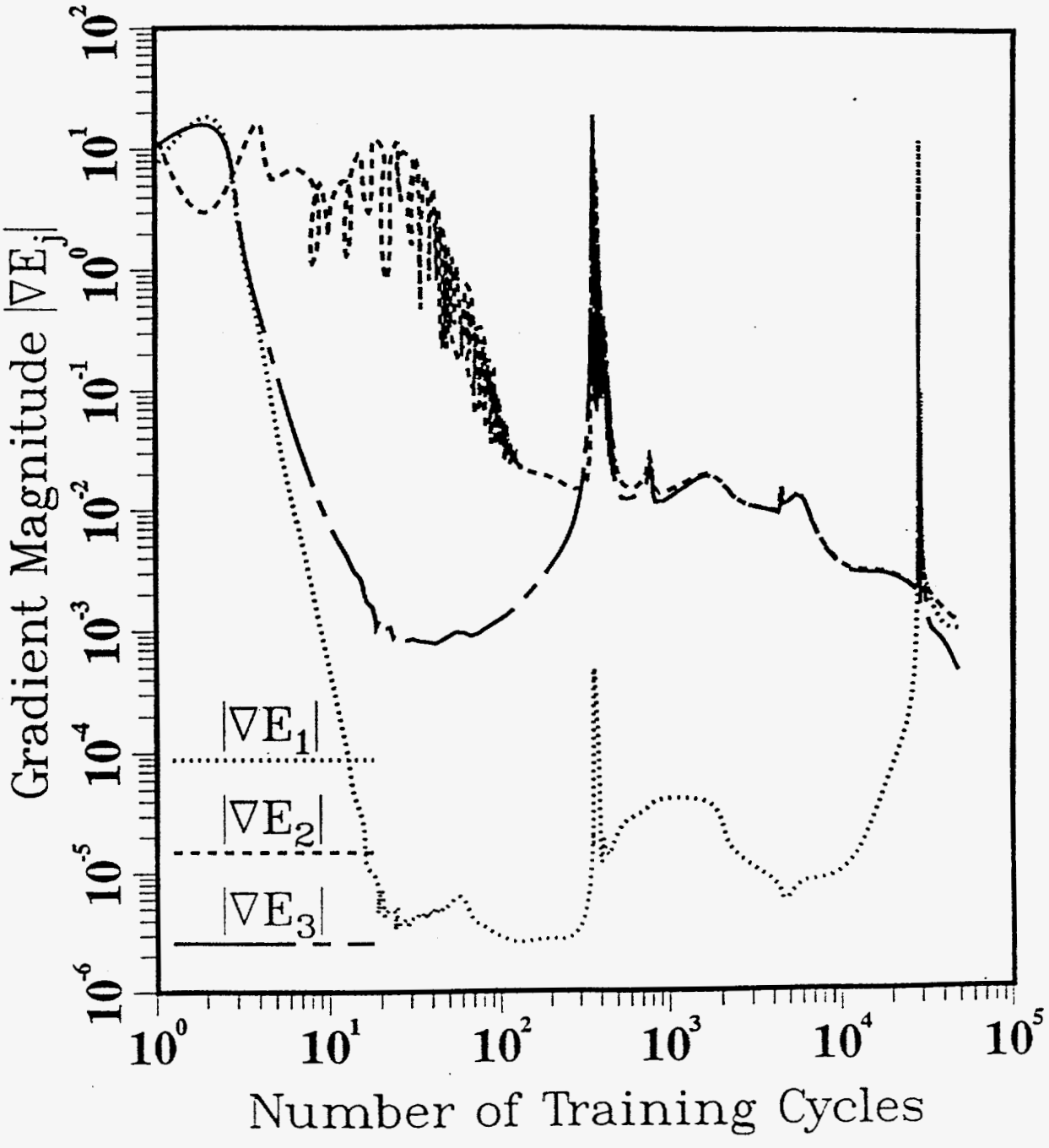

Fig. 4. Typical Activation Level of the Three Output Units for a Given Pattern When Premature Saturation Occurs in Units 1 and 3