Avoiding False Local Minima by Proper Initialization of Connections

Lodewyk F. A. Wessels and Etienne Barnard

Abstract— **The training of neural-net classifiers is often hampered by the occurrence of local minima, which results in the attainment of inferior classification performance. It has been shown [1] that the occurrence of local minima in the criterion function can often be related to specific patterns of defects in the classifier. In particular, three main causes for local minima were identified. Such an understanding of the physical correlates of local minima is important, since it suggests sensible ways of choosing the weights from which the training process is initiated. A new method of initialization is introduced which is shown to decrease the probability of local minima occurring on various test problems.**

I. Introduction

It is well known that the training of the back-propagation (BP) classifier [2] can be viewed as the optimization of a criterion function with respect to a set of parameters—the weights. For efficiency, a local optimization technique is almost always employed for this purpose [3]. Consequently only local minima can be converged upon. If such a minimum happens to be the *global* minimum of the criterion function, the result is a properly trained classifier, but under other circumstances an inferior classifier results.

In realistic cases, the BP criterion function is characterized by a large number of local minima with values in the vicinity of the best or global minimum. Although these minima are not the best possible solutions to the problem, the resulting classifier is close to optimal, deeming the search for a better minimum unjustified. Whenever the optimization process converges upon one of these local minima,[1] the training process is regarded as successful. There are, however, local minima which result in poorly trained classifiers, incapable of close-to-optimal classification performance. Such local minima are referred to as *false* local minima.

The physical states of a network corresponding to false local minima have already been identified and classified by the authors [1]. This knowledge concerning the causes for local minima suggests sensible ways of choosing the weights from which the training process is initiated. In what follows, a new method of initialization is introduced which employs measures aimed at avoiding the most discernible physical states which correlate with local minima.

Our attention will be limited to three-layered networks throughout. Such networks have been shown to be complete in an important sense [4], and are more amenable to conceptual analysis than arbitrarily connected feedforward networks.

II. Avoiding Local Minima

To date, the choice of starting points of the optimization process (which is determined by the weights of the net at the onset of the training process) has not received much attention. Initializing the net with small[2] random weights is the only commonly employed rule concerning the choice of starting weights. The motivation for starting from small weights is that large absolute values of weights cause hidden nodes to be highly active or inactive for all training samples, and thus insensitive to the training process. The randomness is introduced in order to "break symmetry" (preventing nodes from adopting similar functions) [2]. We choose to approach the problem of local minima by initializing the weights appropriately rather than by adaptation of the optimization process [5] or the criterion function [6]. To a large extent these improvements are all independent of one another, and it is likely that an optimal scheme will involve all three. However, since the issue of initialization has received the least attention to date, that will be the sole focus of the current work.

A. The Conventional Method of Initialization

As was mentioned, certain parameters of the conventional method of initialization are not clearly defined. A closer investigation into this method is regarded as an essential step prior to the development of a completely new method. Since the weights are treated as random variables in the conventional method, such an investigation will logically include a mathematical description and discussion of the statistics concerning these random variables.

The input to a hidden node is given by

$$y = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_n x_n, \tag{1}$$

with x_i the activity of the ith neuron in the layer prior to the neuron under consideration, and w_i the weight from that neuron. From (1) the mean of the input to a hidden node is given by

$$E\{y_j\} = E\left\{\sum_{i=0}^{n} \omega_{ji} x_i\right\}$$

Manuscript received June 11, 1991; revised December 10, 1991.

L. F. A. Wessels is with the Division for Materials Science and Technology, CSIR, P. O. Box 395, Pretoria 0001, South Africa.

E. Barnard is with the Department of Electronic and Computer Engineering, University of Pretoria, Pretoria 0001, South Africa.

IEEE Log Number 9106796.

[1] Positions in weight space from where no direction in which the value of the criterion function decreases can be found—because either none exists or limited machine precision obscures them—are regarded as local minima.

[2] No clear definition of "small" seems to exist.

$$= \sum_{i=0}^{n} E\{\omega_{ji}\}E\{x_i\}$$
$$= 0. \tag{2}$$

The second step in (2) is the result of the fact that the weights are independent of the input features. The mean of the input to a hidden node is 0 since the weights leading to a hidden node are regarded as zero-mean-random variables. As a consequence of the result in (2) one is forced to consider the second-order statistics of this problem. Thus, the variance of y is given by

$$\sigma_y^2 = E\left\{(y_j)^2\right\} - E^2\{(y_j)\}$$
$$= E\left\{\left(\sum_{i=0}^{n} \omega_{ji}x_i\right)^2\right\} \quad \text{(from (2))}$$
$$= \sum_{i,k=0}^{n} E\{(\omega_{ji}\omega_{jk}x_ix_k)\}. \tag{3}$$

Since the different weights leading to a hidden node as well as the weights and the input features are independent, (3) simplifies to

$$\sigma_y^2 = \sum_{i=0}^{n} E\left\{(\omega_{ji})^2\right\}E\left\{(x_i)^2\right\} \quad \forall\, j. \tag{4}$$

We assume that the features of the training samples are normalized to lie within the interval $[0; 1]$. The random variables representing the input features are therefore assumed to be uniformly distributed within that interval. Thus,

$$E\left\{(x_i)^2\right\} = \frac{1}{3}. \tag{5}$$

If an input-to-hidden weight is also regarded as a random variable with 0 mean, uniformly distributed in the interval $[-a; a]$, the standard deviation of the input to a hidden node is given (from (4) and (5)) by

$$\sigma_y = \sqrt{N}\,\frac{a}{3}. \tag{6}$$

where N denotes the number of weights leading to a particular node. Exactly the same expression describes the variance of the input to an output node if the activities of the hidden nodes are also assumed to be uniformly distributed.

By generating weights within the range $[-3A/\sqrt{N}; 3A/\sqrt{N}]$, the input to a hidden or output node is a random variable with a standard deviation of A, independent of the number of weights connected to that node. Fig. 1 depicts the average classification performance over 50 runs (with different sets of random initial weights) of the training process as a function of the standard deviation, A. (Details of the data involved—data set 1—as well as the training procedure are provided in Section III). These results indicate that the conventional method performs close to optimally when the value of A is approximately 1. We will consequently choose the weights to occupy the range $\left[-3/\sqrt{N}; 3/\sqrt{N}\right]$ whenever employing the conventional method.

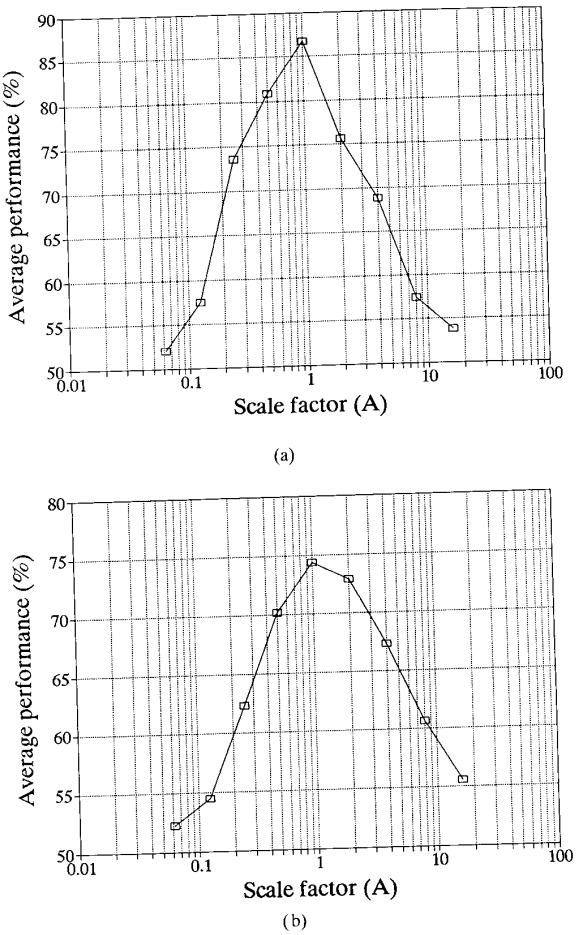


Fig. 1. Effect of the standard deviation of the input to a node on the classification performation of the net: (a) three hidden nodes; (b) four hidden nodes.

B. A Refined Method of Initialization

The most prominent (and most frequently encountered) classes of physical correlates of false local minima are [1]:

- Type 1: Stray hidden nodes, i.e., hidden nodes of which the decision boundaries have been moved out of the region in sample space where the training samples are present, and are either highly active or inactive for all training samples.
- Type 2: Hidden nodes duplicating function.
- Type 3: Hidden nodes arranged such that all are inactive in a certain region of sample space, thus creating a "dead region."

The training processes which ended up in these false local minima started from small randomly generated sets of weights—the conventional initialization routine. The fact that type 1 and type 2 represent two of the most discernible classes of these false local minima is certainly sufficient reason to believe that this initialization routine succeeded neither in breaking symmetry (preventing duplication of function) nor in preventing hidden nodes from straying.

We are therefore led to conclude that a method of initialization intent on curing the problem of local minima should initialize the weights in such a fashion that the following conditions are met: 1) the decision boundaries of the hidden nodes should be positioned well within the region occupied by the training samples; 2) the orientations of the decision boundaries of the hidden nodes have to be as varied as possible; and 3) every part of the sample region needs to have at least one hidden node which is active for samples occupying that region.

In a Cartesian coordinate system the n-dimensional decision hyperplane associated with a hidden node is mathematically described by the weights, $\omega_0, \cdots, \omega_n$, leading to that node. For the purposes of this work, we will switch to spherical coordinates. By defining an n-dimensional vector with a length parameter, r, and $(n-1)$ angles $(\theta_1, \cdots, \theta_{(n-1)})$, one can uniquely refer to the hyperplane perpendicular to such a vector, positioned at its endpoint. See Fig. 2. (Note that the origin has been shifted to (0.5; 0.5).) This change of coordinate systems is suggested by the nature of conditions 1 and 2 to be built into the improved method of initialization: the length, r, controls the distance that the decision boundary is removed from the center of the area where the training samples are situated, whereas the different angles present a natural way of controlling the orientations of the hyperplanes.

The angle variables are chosen such that the decision hyperplanes are uniformly oriented throughout the feature space, at the same time maximizing the difference in orientation between the various hidden nodes' hyperplanes. In two dimensions the values of the angle variables are given by

$$\left\{\theta^{(j)}\right\}_1^d = \begin{cases} \left(\frac{\pi}{d} \cdot j\right) & j \text{ odd} \\ \left(\frac{\pi}{d} \cdot j + \pi\right) & j \text{ even.} \end{cases} \tag{7}$$

Note that the angle variables are limited to the range $[0, \pi]$ in order to prevent hyperplanes from adopting parallel orientations. The offset of π added to the values of the even-numbered nodes compensates for this limitation, ensuring that the hyperplanes occupy both halves of the feature space.

In determining the values of r, two constraints are met: the hidden nodes' hyperplanes are positioned well within the sample region, simultaneously maximizing the variance in their positioning. It was decided to choose the r parameters as follows:

$$r^{(0)} = 0.5$$
$$r^{(j)} \leftarrow \frac{1}{(j+1)} \cdot r^{(0)}$$
$$\left(\text{i.e., } r^{(1)} = 0.25, r^{(2)} = 0.167, r^{(3)} = 0.125 \text{ etc.}\right) \tag{8}$$

Thus far we have only specified the position of the decision boundary, i.e., where y in (1) equals 0. It remains to choose 1) the direction in which this function increases (since both y and $-y$ have the same decision boundary) and 2) the slope at which y increases/decreases (since y and αy have the same decision boundary). Choice 2 allows us to manipulate the initial sensitivity of the net: an increase in the value of α results in a hidden node of which the sigmoidal transfer function is characteristically steep. Since such a situation results in small values of the gradient with respect to the weights associated with such a hidden node, the training process has very little effect on that hidden node's weights, resulting in a "rigid" decision boundary.

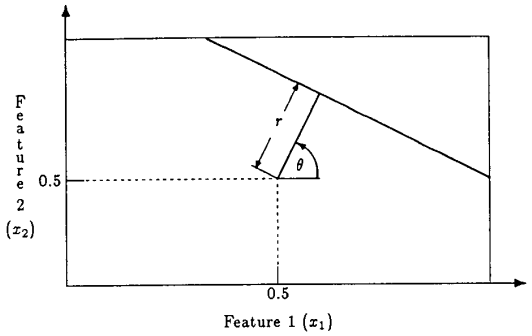


Fig. 2. Parameterization of a decision boundary in spherical coordinates.

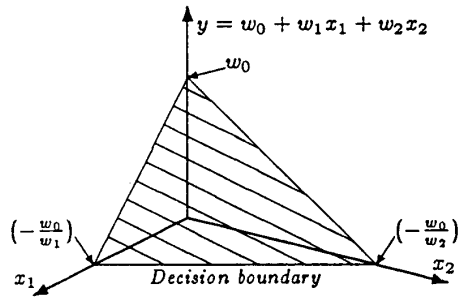


Fig. 3. Relationship between a hyperplane in augmented feature space and its corresponding decision boundary in nonaugmented feature space. The relationships between the intercepts of the plane and the weights are also shown.

Choice 1, on the other hand, is employed to position the active region of the hidden nodes' decision hyperplanes in such a way that the development of a dead region is prevented. This is accomplished by choosing the threshold weights of the hidden nodes such that the origin of the Cartesian feature space is contained in the active region of all the hidden nodes. See Fig. 3. This, together with the fact that the decision boundaries are quite uniformly spread throughout the feature space, results in effective prevention of the occurrence of dead regions.

It is clear that this approach to hidden-node initialization can straightforwardly be extended to spaces of higher dimensionality, and we have employed such an extension.

Having established an appropriate way of selecting the starting values of the input-to-hidden weights, we can now proceed to the hidden-to-output weights. Since the developed method effectively breaks symmetry by a proper choice of the input-to-hidden weights, different criteria govern the choice of hidden-to-output weights. The amount learned by each hidden node during each iteration of the training process depends to a large extent on the value of the hidden-to-output weights connected to these weights. Very small values of these weights will slow the training process down considerably. Excessively

large values of these weights, on the other hand, result in large error signals being fed back from the output nodes during training. Such large error signals lead to drastic adjustments to the input-to-hidden weights and consequently to hidden nodes which are insensitive with regard to the training process. Choosing all the hidden-to-output weights equal therefore seems to be a good option, since this ensures that the training process commences from a situation where the hidden nodes are all learning approximately the same amount on every iteration. Such a situation is preferable since it will prevent the hidden-to-output weights from "disguising" the positive aspects of the way in which the input-to-hidden weights were initialized.

III. Experimental Results

To determine the validity of the initialization process and to obtain suitable values for the parameters which were not defined in the discussion of Section II, we conducted various tests with this method of initialization as well as the conventional method of initialization. In all cases we employed a three-layered back-propagation network with the number of input nodes one more than the number of input features, the number of output nodes equal to the number of classes, and the number of hidden nodes determined as described below. Training was performed using conjugate-gradient optimization [7], with target output vectors of the form $(0, \cdots, 0, 1, 0, \cdots, 0)$ the 1 being in the nth position for an input in class n. The criterion function employed was the standard sum-of-squares described in [2].

Three data sets are involved in the experiments conducted in this work. Data set 1, which is depicted in Fig. 4(a), is a two-dimensional data set. It consists of 104 samples comprising two classes. Another artificially generated two-dimensional data set, involving 383 samples from three classes, comprises data set 2. See Fig. 4(b). Data set 3 is a five-dimensional data set consisting of 843 samples comprising three classes. The samples belonging to class 1 were uniformly distributed, whereas the samples in classes 2 and 3 assumed unimodal and bimodal Gaussian distributions respectively. The first two classes were positioned between the two clusters constituting class 3. The classes were also positioned such that no separation was possible in the fourth and fifth dimensions. Fig. 4(c) is a representation of the two-dimensional section of this five-dimensional data set—the x_2x_3 plane. The Bayesian error rate on this data set was determined using Monte Carlo integration and was found to be 4.97% with a standard deviation of 0.082%.

The smallest number of hidden nodes to produce satisfactory[3] classification performance was determined for the different data sets. These were found to be three or four hidden nodes in the case of test problem 1, six hidden nodes for data set 2, and ten hidden nodes for test problem 3.

The experiments which are reported in this section were conducted as follows: the net was initialized in the relevant manner[4] and trained on one of the test problems. The classification performance of the trained net was then determined using the training set as a test set. Generalization is usually also taken into account when evaluating the performance of a classifier. Although not a completely true measure of general-

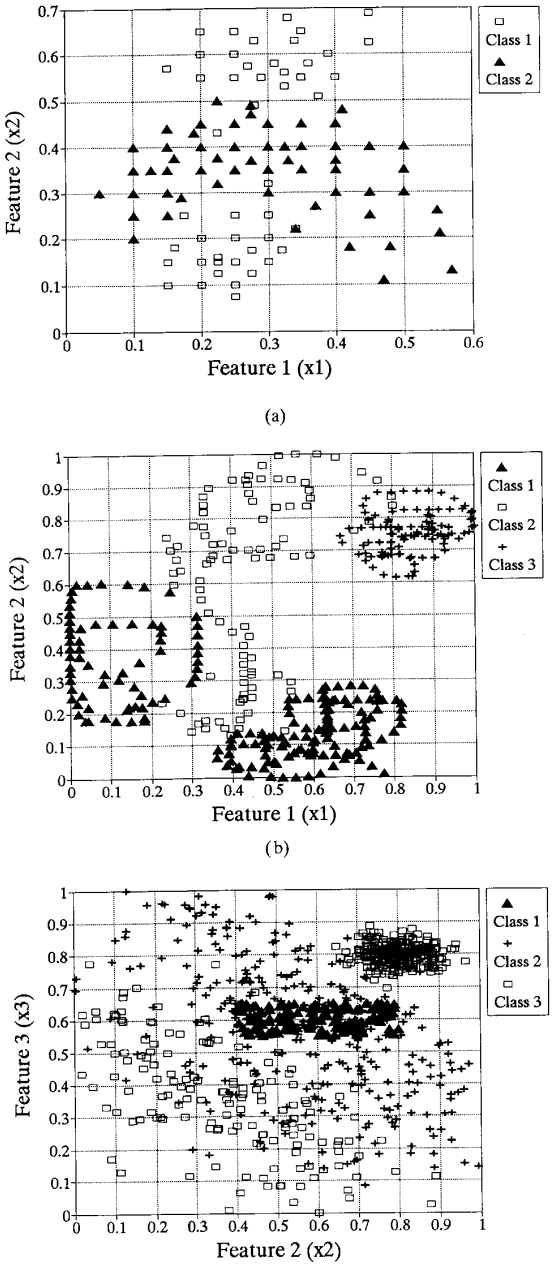


Fig. 4. (a) Data set 1. (b) Data set 2. (c) A two-dimensional section through data set 3, providing limited though visually representable insight into this data set.

[3] No improvement in classification performance is achieved by adding hidden neurons.

[4] Employing either the conventional initialization routine or the new method of initialization, depending on the specific phenomenon investigated in the particular experiment.

ization [8], the value of the criterion function was regarded as a sufficient indication. In order to obtain representative statistical data, the process was repeated 50 times, each time starting from a different set of weights. After the 50 runs were completed, the average classification performance of the net during these 50 runs was calculated.

In further discussions a single training process will be referred to as a *trial* and a collection of 50 of these trials will be called an *experiment.*

A. Optimal Input-to-Hidden Weights

For random initial weights it is customary to scale the weights leading to a node to some predetermined variance. However, it is not clear that the same arguments apply to our new method of initialization. We therefore investigated the effect of scaling the input-to-hidden weights, keeping all initial hidden-to-output weights fixed at 0.25. To do this, the set of weights constituting a starting point was scaled such that the sum of the square values of the weights leading to a hidden node ranged from a value of 3 up to 12 800 (i.e., $S \in [3, 12\,800]$, where S is the sum of squares). The results on test problems 1 and 2 are presented in Fig. 5.

It was found that the values of S for which best performance was achieved are $S = 100$ for test problem 1, regardless of whether three or four hidden nodes were employed; $S = 200$ in the case of test problem 2; and $S = 800$ for the five-dimensional problem.

The results achieved with both the new and the conventional method are presented in Fig. 6.

B. Optimal Hidden-to-Output Weights

The reader will recall that, whereas the scaling of the input-to-hidden weights was investigated in the previous section, the values of the hidden-to-output weights were fixed at 0.25. Having optimized the choice of the input-to-hidden weights for this value of the hidden-to-output weights, we can keep the input-to-hidden weights fixed at these optimal values, and in turn optimize the value assigned to the hidden-to-output weights. The average classification performance as a function of the value assigned to the hidden-to-output weights for test problems 2 and 3 is presented in Fig. 7. The graphs indicate that the classification performance is clearly insensitive to the values of the hidden-to-output weights, as long as this value is less than 1. For values in excess of 1, there is a dramatic decline in the classification performance.

The fact that virtually no increase in the classification performance results for hidden-to-output weights smaller than 1 justifies our initial choice of 0.25.

IV. Discussion

The optimal values of S, as determined experimentally, differ considerably from the value of the variance ($\sigma^2 = 3$) previously assigned to the random variable representing the weights leading to a hidden node. The motivation for assigning such a value to the variance was to prevent the outputs of the hidden nodes from saturating from large inputs caused by large input-to-hidden weights. This goal was achieved.

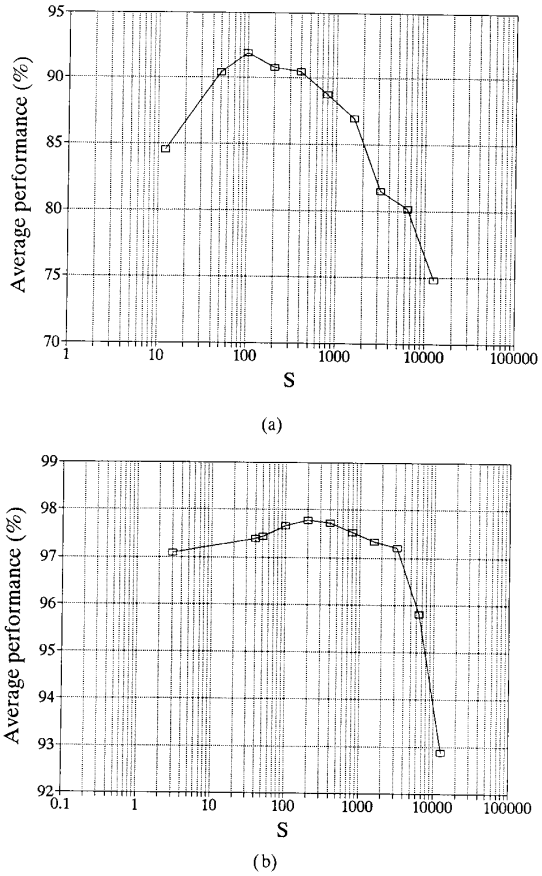
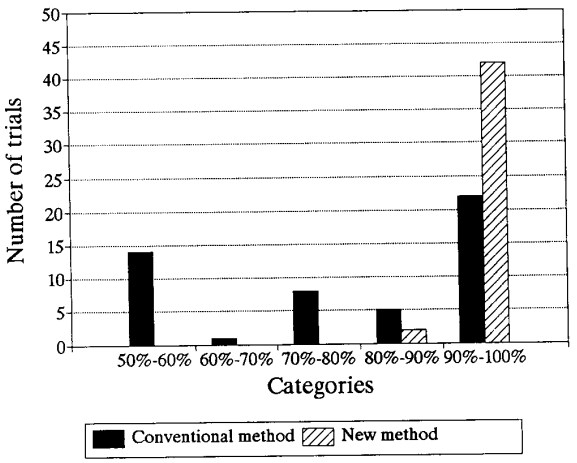
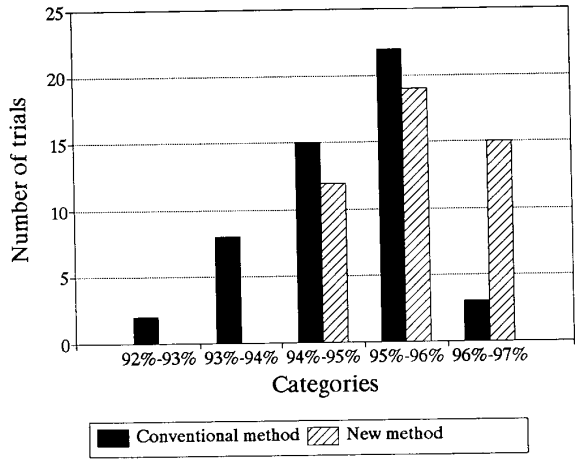


Fig. 5. Effect of the sum of squares of the input-to-hidden weights (S) on the average performance of a net initialized with the improved method. (a) Three hidden nodes (test problem 1). (b) Data set 2.

Such nodes, i.e., nodes with gradual transition regions, are desirable in the sense that they are sensitive to the training process. However, such mildly sloped transfer functions have two drawbacks. First, it has been observed quite frequently that such decision boundaries tend to be moved through considerable distances during the training process before settling into a stable position. These include instances in which the decision boundary was pushed completely out of the sample region—a type 1 local minimum. This is related to the second negative aspect of decision boundaries with gradual transitions, namely that they tend to approximate a linear transfer function. Suppose that a net is initialized with a set of weights such that the hidden nodes are characterized by approximately linear transfer functions. The combination of the *linear* transfer function of one hidden node with that of another produces only a different *linear* function, resulting in a linear decision boundary at the output nodes. Clearly, such a decision boundary could have been constructed by a single hidden node. Since the specific discriminating function of each of the two hidden nodes is obscured by their combined functioning, the following are possible consequences: one hidden node fulfilling the necessary function whereas the other one is forced out of the sample region or both hidden nodes

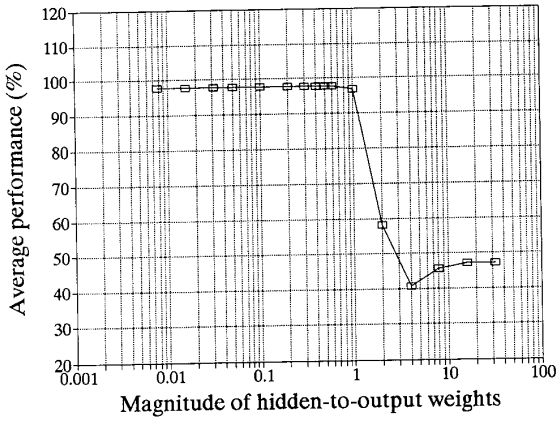


(a)

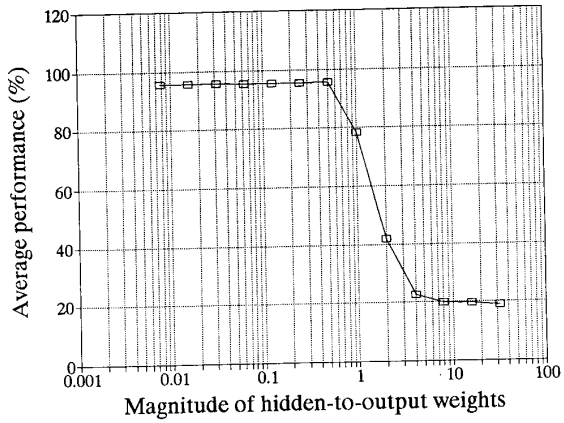


(b)

Fig. 6. Outcomes of the experiments initialized according to the new and conventional methods. (a) Test set 1. (4 hiddens; $S = 100$). (b) Test set 3. (10 hiddens; $S = 800$). The amplitude of each bar denotes the number of times the training process produced a classifier with a classification performance within the range represented by the bar.



(a)



(b)

Fig. 7. The average classification performance of the net as a function of the value assigned to the hidden-to-output weights: (a) test problem 2; (b) test problem 3.

assuming approximately the same function. This linearity tends to occur only during the initial stages of the training process, since the transitions become steeper as the weights grow larger. However, by the time the training process reaches a stage where the values of the weights increase, resulting in more abrupt transition regions which are far from linear, the damage has already been done (in one of the above two senses) and a false local minimum results.

"Insensitive," or "rigid," decision boundaries represent the other undesirable extreme situation related to the scaling of the input-to-hidden weights. Inspection of the transition regions of the starting weight producing optimal classification performance reveals that the width of the transition regions is such that these regions occupy approximately 20% of the range of a single variable in feature space. Owing to the superior results associated with such a situation, the conclusion can be drawn that this situation represents a balance between rigid and approximately linear transfer functions; sufficient training samples populate that region of sample space constituting the transition region, resulting in a non-zero value of the derivative of the criterion function with respect to the weights for these samples. Some adjustment to the decision boundaries is thereby permitted. However, since drastic movements of the boundaries are not permitted (owing to a sufficient degree of steepness in the transitions) the learning process is prevented from nullifying those characteristics that were deliberately incorporated into the initialization routine. Thus, by initiating the training process with weights with the indicated variance, the "creativity" of the net was limited to a certain extent. However, the possible network configurations which were consequently less likely were exactly those that mainly lead to false local minima.

Large hidden-to-output weights amplify the error signals which are fed back through the net from the output nodes. The input-to-hidden weights, which are adjusted according to these error signals, are therefore perturbed drastically. This results in extensive shifts in the positions of the decision boundaries of these nodes. This phenomenon was observed quite frequently

when the value of the hidden-to-output weights exceeded 1. Experimental evidence indicates that such shifts result in stray hidden nodes in the majority of cases.

Another noteworthy phenomenon is the fact that the optimal performance of the different test problems tended to occur at different values of S. However, in all the experiments satisfactory performance was achieved for values of S ranging from 100 to 800. Since this seems to be one of the parameters of which the exact value is not crucial to the success of the new method, we recommend that a value of $S \sim 200$ be used.

It has to be borne in mind that problems different from those investigated in this work might require different values for the various parameters or even different methods of initialization to effectively counter the problem of false local minima.

We conclude by commenting on the merits of this new method of initialization:

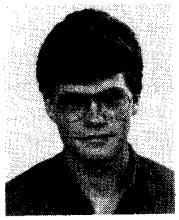
- The average classification performances attained with the conventional method of initialization were improved from 86.7% to 94.5%, from 96.7% to 97.8%, and from 94.8% to 95.7% for test problems 1, 2, and 3 respectively. This corresponds to reductions in the average error rate of the classifier of 59%, 33%, and 16%.
- Even more important than the improvement in the average error rate is the fact that the occurrence of false local minima during the training processes has been reduced to a large extent. This is indicated by the results depicted in Fig. 6. The reduction in the occurrence of local minima establishes itself in the notable shift in the histograms away from the lower classification performances toward those percentages representing well-trained classifiers. For instance, in the case of test problem 1, classification performances worse than 90% were regarded as false local minima. The result of the new method on this problem, depicted in Fig. 6(a), shows a reduction of 52% in the number of false local minima that were encountered. Although not as dramatic as for this test problem, the experiments on the other test problems also indicated a notable decrease in the occurrence of false local minima.

V. Conclusion

By initializing weights to avoid the circumstances associated with local minima, we have been able to obtain improved performance on three different test problems. Such simulations cannot be conclusive, owing to the statistical nature of this problem domain. We therefore regard these results as *suggestive*. It is hoped that further experimentation along these lines will shed further light on issues such as the appropriateness of the various selected parameters, and thereby provide additional evidence for the validity of this method.

References

[1] L. F. A. Wessels, E. Barnard and E. van Rooyen, "The physical correlates of local minima," in *Proc. Int. Neural Network Conf.* (Paris), July 1990.
[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*. Cambridge MA: MIT Press, 1986, ch. 8, pp. 318–362.
[3] R. L. Watrous, "Learning algorithms for connectionist networks: applied gradient methods of non-linear optimization," in *Proc. Int. Conf. Neural Networks* (San Diego, CA), July 1987, pp. II-619–II-627.
[4] M. Stinchcombe and H. White, "Universal approximation using feed-forward networks with non-sigmoid hidden layer activation functions," in *Proc. Int. Conf. Neural Networks* (Washington DC), June 1989, pp. I-613–I-618.
[5] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science*, vol. 9, pp. 147–169, 1985.
[6] S. A. Solla, E. Levin, and M. Fleisher, "Accelerated learning in layered neural networks," Tech. Rep., AT&T Bell Laboratories, 1988.
[7] M. J. D. Powell, "Restart procedures for the conjugate-gradient method," *Mathematical Programming*, vol. 12, pp. 241–254, 1977.
[8] E. Barnard and D. Casasent, "A comparison between criterion functions for linear classifiers with an application to neural nets," *IEEE Trans. Syst., Man, Cybern.*, to be published.



Lodewyk F. A. Wessels received the B.Eng. and M.Eng. degrees in electronics from the University of Pretoria, South Africa.

He is currently a Project Engineer at the Division of Materials Science and Technology, CSIR, South Africa. His research interests include neural networks with application to modeling, control, and speech recognition.

Etienne Barnard received the B.Eng degree in electronics from the University of Pretoria, South Africa, the B.Sc. (Hons) and M.Sc. degrees in physics from Witwatersrand University, South Africa, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA.

He is currently Assistant Professor of Electronics and Computer Engineering at the University of Pretoria. His research interests include pattern recognition with application to speech and image understanding, and computer-generated holography.