J M Bishop
10/9/91.

South African Computer Journal
Number 5
September 1991

Suid-Afrikaanse Rekenaartydskrif
Nommer 5
September 1991

Computer Science and Information Systems

Rekenaarwetenskap en Inligtingstelsels

The South African Computer Journal

An official publication of the South African Computer Society and the South African Institute of Computer Scientists

Die Suid-Afrikaanse Rekenaartydskrif

'n Amptelike publikasie van die Suid-Afrikaanse Rekenaarvereniging en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes

Subscriptions

	Annual	Single copy
Southern Africa:	R32-00	R12-00
Elsewhere:	\$32-00	\$12-00

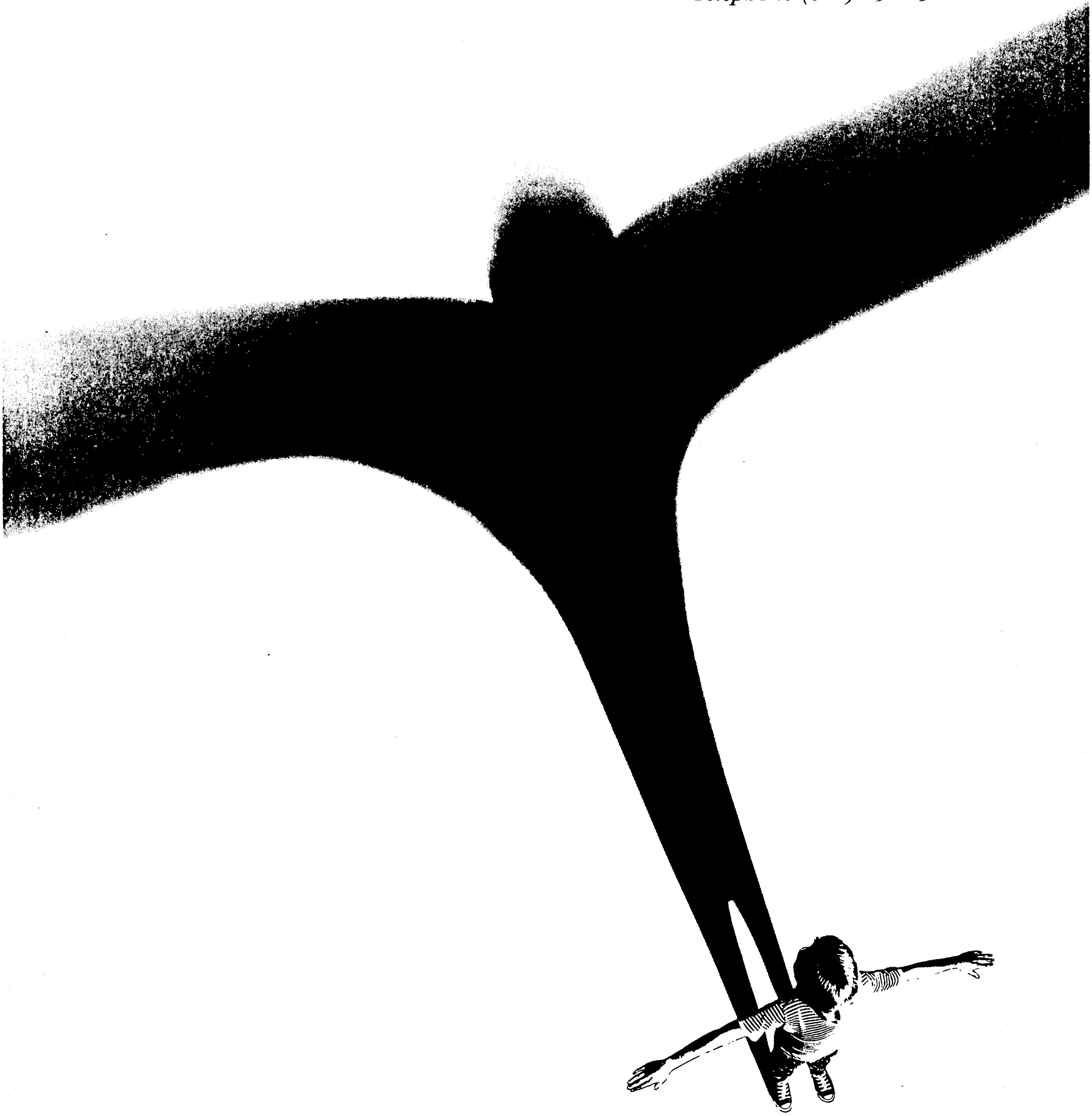
to be sent to:
Computer Society of South Africa
Box 1714 Halfway House 1685

Computer Science Department, University of Pretoria, Pretoria 0002
Phone: (012) 420-2843 Fax: (int)+(012)+43-6454
Sigma Press

Regardless of the size of your computer system it is a strategic tool. We are committed to helping you use it most effectively. All it needs is a little imagination.

UNIDATA

Telephone (012) 45-1152



THERE'S A GIANT IN EVERY ONE OF US

YOUNG & RUBICAM 108113

UNIVERSITY OF PRETORIA
COMPUTER SCIENCE
MSc. BURSARIES
FOR STUDENTS FROM AFRICA

Universiteit van Pretoria

The Department of Computer Science at Pretoria University is inviting applications for two annually awarded bursaries for MSc study. These bursaries are aimed at outstanding students who have a Computer Science Honours degree (or equivalent qualification). As far as possible, one such bursary will be awarded to a South African student, while the other will be awarded to a student from another African state. The hope is to thereby promote interaction between South African computer scientists and computer scientists elsewhere in Africa.

The bursaries are funded by the Department's contract research work and will amount to a minimum of R20000 per year. They are intended to cover costs for tuition, housing, transport, books, and an adequate allowance for other living expenses. Successful candidates will be required to work for 20 hours per week as part of the Computer Science Department's contract research team. The work assigned to each candidate will be aimed at providing him with practical experience in his field of interest.

Study fields for which bursaries will be considered include:

- Computer Networking
- Distributed Systems
- Software Engineering
- Artificial Intelligence
- Intelligent Computer Aided Instruction
- Machine Aided Translation

Students should aim to complete the dissertation that is required for an MSc within two years of study (i.e. one year of full time study). The dissertation should preferably be written in English but it may also be written in Afrikaans.

The continuation of the bursary and adjustments to the allowance will be decided upon annually and will depend on progress. It is a condition of the bursary that, after completing his/her studies, the student should return to his/her country of origin for the same number of years for which the bursary was held.

Applicants may request application forms from:

MSc. Bursaries
The Computer Science Department
University of Pretoria
Hillcrest
Pretoria
0083
South Africa

Editor's Notes

It is with sincere gratitude that SACJ takes leave of Dr Peter Lay who, until recently, was the assistant editor dealing with Information Systems. He has left academia for what sounds like a more gentle lifestyle. (He has gone farming!) Under Peter's stewardship the number of high-quality IS papers in SACJ grew steadily. In general, IS papers tend to be accessible and relevant to a wide spectrum of computer professionals, and the quality of IS papers that have been appearing in SACJ has significantly contributed to the increased interest being shown in the journal by the local computer industry. If this growth in interest is to be sustained, it is urgent and important to find a suitable replacement assistant editor. The ideal candidate should not only be respected as an academic by his peers, but should also be disposed to enthusiastically promote SACJ in the private sector. Since a shortlist of candidates is currently being compiled, I would like issue a general appeal for names that might be included on it. Please contact me urgently if you would like to be considered for the job, or if you would like to nominate someone that you consider to be particularly suitable.

My three year term of office as editor expires in October. I have always considered it a great privilege to hold this position, and as a result, I felt honoured when the SAICS executive committee requested that I stay on for a further term. Nevertheless, I initially declined the request on the grounds that the time-demands of the job were significantly eroding my ability to fulfil other duties. Particularly demanding has been the task of seeing to the typesetting of the various contributions - either by doing it myself, or by ensuring that it is adequately done by someone else. Recently, however, Prof G de V Smit (Riël Smit) at UCT has offered to assume the role of production editor. This generous offer so much changes the complexion of what is being asked of me that I am now both willing and honoured to continue as editor for another term. I am very grateful to Riël for his offer and I look forward to working with him. In future, authors whose papers have been accepted for publication will be asked to liaise directly with him regarding the precise form in which the final contribution should be submitted.

The next issue of SACJ will consist largely of a selection of papers that were presented at the 6th South African Computer symposium. The selection will be based on comments from the referees who, at the time, were asked to adjudicate the papers in terms of their appropriateness for both the conference as well as for SACJ publication. Papers which, in the opinion of one or more referees, required major revision will have to be resubmitted to SACJ for refereeing purposes. Authors will soon be contact in this regard.

At the time of writing, the updated list of "approved" publications for the first half of 1991 had not yet been released by the relevant authorities. For the sake of past, present and future contributors I sincerely hope that SACJ will be on the list when it eventually comes out. However, I have become increasingly aware that there is a real danger of laying too much store on papers published in so-called approved journals as a basis for evaluating and rewarding research. I hope to expand more fully on this theme in a future edition of SACJ. Keep watching this space!

Derrick Kourie
Editor

This SACJ issue is sponsored by
a generous donation from
UNIDATA

The Physical Correlates of Local Minima

L.F.A.Wessels [1], E Barnard [2], E van Rooyen [1]

[1] *Carl and Emily Fuchs Institute for Micro Electronics, University of Pretoria, Pretoria 0001.*

[2] *Department of Electronic and Computer Engineering, University of Pretoria, Pretoria 0001.*

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. We study the problem of local minima in order to devise means of alleviating it. In order to establish a better understanding of the problem, the nature of the physical states of neural nets stuck in local minima, is investigated. We show that the occurrence of a local minimum in the criterion function can often be related to specific patterns of defects in the classifier. In particular, three main causes for local minima are 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.

Keywords: *neural networks, classifier, backpropagation, training, criterion function, decision boundaries, local minima.*

Computing Review Categories: *I.5.2*

Received November 1990, Accepted July 1991

1 Introduction

The backpropagation (BP) classifier [1] has during recent years established itself as a very efficient and robust neural net classifier[2] [3] [4]. A typical architecture of this network, when utilized as a classifier, is depicted in Figure 1.

As shown in this figure, the classifier consists of three layers of neurons: the 'input' layer (i.e. the first layer), the 'hidden' (second) layer and the 'output' or third layer. This is a *feedforward* network, in which only connections between adjacent layers are allowed. No recurrent connections are allowed, which determines that all neurons can only be influenced by neurons occupying an earlier layer. In this network, the activities of the input neurons represent the features, x_i, of the input sample, $\vec{x}$, which has to be classified. (Provision is usually made for an additional 'threshold' node[5].) These input values are then multiplied by w_{ij}, the weights connecting the input nodes to the hidden nodes. The products leading to a particular hidden neuron are then added together. The activation of the the j^{th} hidden node, y_j, is a non-linear function of these sums:

$$y_j = f\{\sum_i (w_{ji}x_i)\}, \tag{1}$$

The non-linear activation function (also referred to as the 'sigmoidal' function) is given by:

$$f(x) = \frac{1}{2}\{1 + tanh(x)\} \tag{2}$$

The procedure to obtain the activities of the third layer neurons is exactly the same, this time involving the second layer activities (y_j) and the hidden-to-output weights, v_{jk}. The output nodes indicate the class into which a certain input sample is classified. When the k^{th} output node is turned on, and all the other turned off, class k is indicated.

The construction of a BP classifier involves a training set and a criterion function. Consider a network consisting of I input nodes, J hidden nodes and K output nodes. The training set consists of P samples, $\vec{x}_p$, with their associated target vectors, $\vec{t}_p$. The target vectors specify the class to which each sample belongs, e.g.: if the p^{th} sample belongs to the k^{th} class, the p^{th} target vector consists of zeros in all the positions except the k^{th},which contains a '1'. The different features comprising a training sample, are usually scaled to lie in the interval $[0, 1]$. The samples are presented at the input layer of the network upon which the network produces its own output by feeding the input forward through the different layers of weights and nodes, until it reaches the output layer. The square of the difference between the output, $\vec{z}_p$, generated by the network on the presentation of the p^{th} pattern, and the p^{th} target vector, constitutes the contribution of the p^{th} pattern to the criterion function:

$$E_p = \frac{1}{2}\sum_{k=1}^{K}(t_{pk} - z_{pk})^2 \tag{3}$$

This process is repeated for every sample in the training set. The total measure of error (value of criterion function) is then given by

$$E = \sum_{p=1}^{P} E_p \tag{4}$$

It is well known that the training of the BP - classifier can be viewed as the optimization of the criterion function (given in 4) with respect to a set of parameters - the weights. For efficiency, a local optimization technique is almost always employed for this purpose.[6] (We used the conjugate-gradient optimization technique.) Consequently only local minima can be converged upon - due to the non-hill climbing nature of the minimizers, they cannot escape from these local minima. (This is the definition of a *local* optimizer [7].) If the 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 [8]. (Just how many such minima exist is an interesting, but unsolved, question. That this number must be large, is suggested by experiment [9] on a speech - derived data set, where 100 separate trials all converged to distinct minima.) 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, the training process will thus be regarded as successful. There are, however, local minima which result in poorly trained classifiers, incapable of close-to-optimal classification performance. Such local minima will be referred to as *false* local minima.

In what follows, the occurrence of false local minima in the criterion function will be related to the configuration and behaviour of the network as such. Special attention will be devoted to explaining the reasons for such occurrences. This will lead to a better understanding of the causes for local minima.

In order to investigate local minima, a two-dimensional test problem was constructed. It consists of 104 training samples comprising two classes. Each sample is represented by two features, which constitute the two-dimensional vector denoting that specific sample. Figure 2 provides a graphic representation of the test set. When constructing a classifier for this problem, the objective is to produce a classifier which

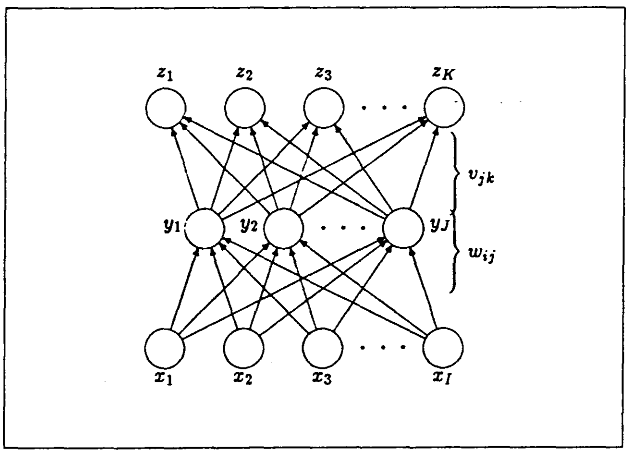


Figure 1: BP classifier.

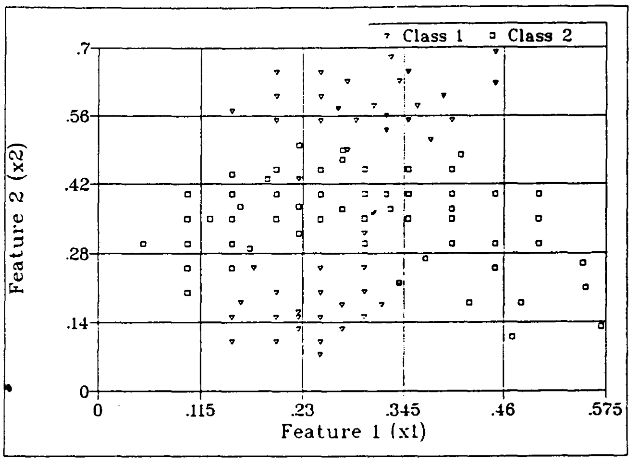


Figure 2: Artificially generated test problem.

separates the two classes optimally. The test problem is regarded sufficient for the purposes of this work since it provides a great variety of examples of false local minima, and since the effects of these local minima can be visually inspected in the two-dimensional feature space. A two-dimensional problem also has the tendency to be less easily separable than higher dimensional problems which makes it even more suitable as a test in classification - consider the following example: suppose a problem consists of two classes to be separated. If these two classes consist of two clusters which are overlapping when presented in two-dimensional space, no complete separation is possible. However, if another dimension is introduced, the possibility exists, that, although the clusters might not be completely separable in the first two dimensions, easy separation is possible due to the third dimension. As the number of dimensions increase, the possibility of separation also increases.

Good solutions to the test problem (i.e. reasonable separation of the various classes, corresponding to approximately 90% or better classification performance on the training set) are attainable with three or more hidden nodes. An example of the decision boundaries[1] constructed in the case of a network employing three hidden nodes is depicted in Figure 3. The linear decision boundaries constructed by the hidden nodes are shown in Figure 4.(The interested reader is referred to [10] for more information on the construction of decision regions.)

Clearly, the net is able to separate the two classes reasonably well. The decision boundaries created by the net with four hidden nodes and optimal performance are virtually the same as in Figures 3 and 4. The fourth hidden node's decision boundary was, however, moved away from the sample region, enabling the

[1]Decision boundaries are defined as those curves in feature space where the outputs of the output(hidden) neurons are equal to 0.5 (or 0.5 ± margin, in which case the boundary will have a width approximately proportional to the size of the margin.) From (1) and (2) the decision boundary for hidden neuron i is given by $\Sigma_j w_{ij} x_j$, i.e. it is linear in feature space.

net to employ it as a threshold to the output nodes. This resulted in a in better positioning of the decision boundaries, and consequently better classification performance.

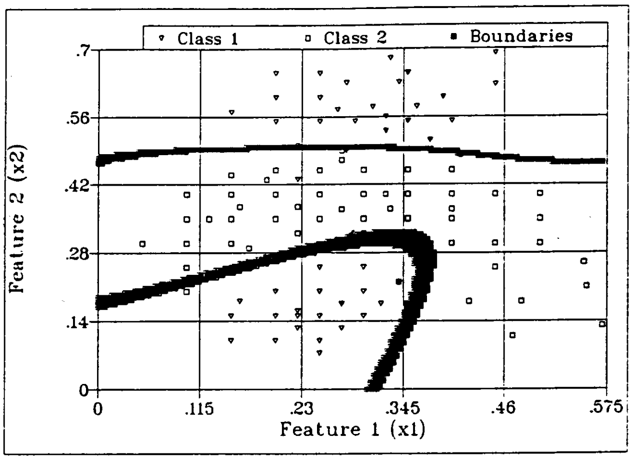


Figure 3: Decision boundaries created by the classifier for the test problem. (3 hiddens)

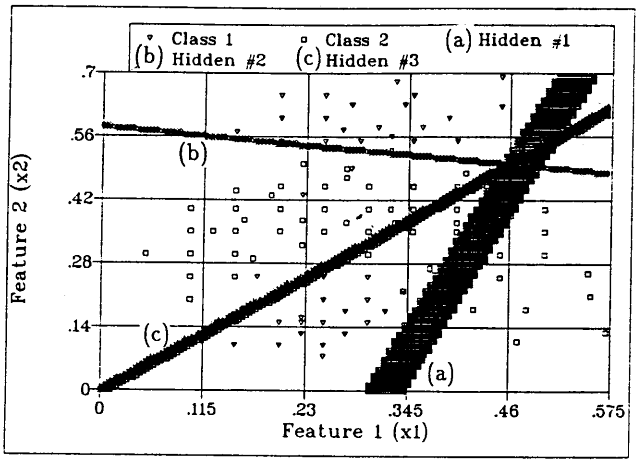


Figure 4: Decision boundaries created by hidden nodes for the test problem (3 hiddens)

2 Classes of false local minima

Neural nets employing first three and then four hidden nodes were repeatedly trained on the test problem, starting the optimization routine from various sets of randomly generated small [2] weights. Whenever the optimizer reached a point in weight space from where no direction along which the value of the criterion function decreases could be found, the training process stopped, indicating that a local minimum had been reached. The classification performance of the classifier with the set of weights indicated by that point, was then compared to the performance of a successfully trained classifier. Whenever it was close to optimal, in this case 90% correct classification or better, the particular local minimum was regarded an appropriate solution. (Such a minimum is therefore close enough to the global minimum). Whenever the classifier performed poorly (worse than 90%), the local minima converged upon were classified as false local minima. Once the occurrence of a false local minimum was established, the resulting network was analyzed.

A classification of the class of false local minimum was made according to the behaviour of the hidden nodes. The kinds of false local minima encountered, will be discussed in detail in the following sections.

2.1 Type 1 : 'Stray' hidden nodes

Figure 5 depicts the linear decision boundaries constructed by the hidden nodes of a neural net which ended up in a typical false local minimum of type 1. Only two of the three hidden nodes' decision boundaries are in the area where the training samples are present, therefore being the only nodes able to perform discriminating tasks. The other hidden node has been pushed out of the sample region, and is turned off for all the training samples. The result is a network trying to perform the classification task with the other two nodes, which results in a poor classification performance of 50%.

It is easy to show that the derivatives of the criterion function with respect to the weights belonging to a node shifted out of the sample region will tend to zero, because the slope of the sigmoid tends to zero for large values of its argument. In addition, the derivatives with respect to the weights of ♯1 and ♯2 are also zero since these neurons are in locally optimal positions. The system is consequently trapped in a local minimum. Since the network performs so poorly, this turns out to be a false local minimum.

[2] Chosen in such a fashion that the sum of the weights leading to a certain hidden node is a random variable having zero mean and variance of 3, resulting in outputs of that node being mostly in the 'active' (unsaturated) region of the sigmoid.

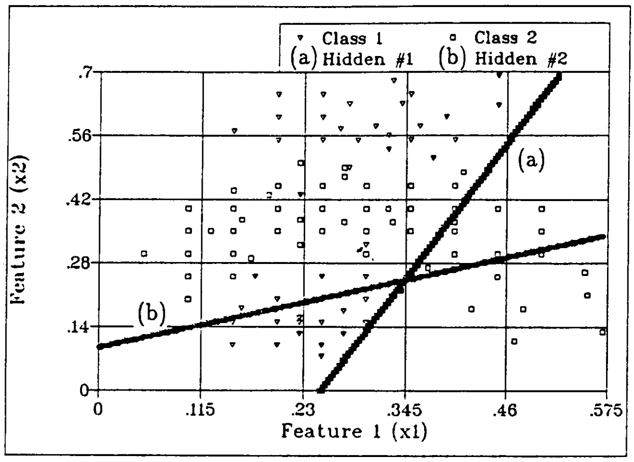


Figure 5: Decision boundaries of hidden nodes with one hidden node straying

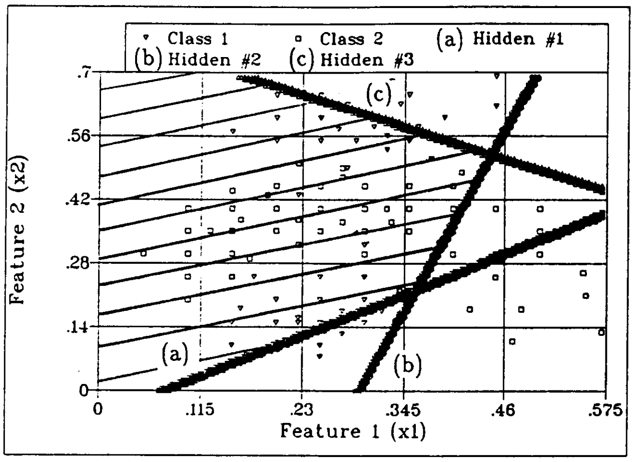


Figure 6: Decision boundaries of hidden nodes with a region where these hiddens are inactive

2.2 Type 2 : Hidden nodes duplicating function

A second type of false local minimum that occurs rather frequently is related to duplication of function by neurons. Thus, with three hidden nodes we often find that two hidden nodes take on positions similar to ♯1 and ♯2 in Figure 5, with the weights of the third hidden neuron (and therefore the position of its decision boundary) exactly equal to those of ♯1 or ♯2.

Since hiddens ♯1 and ♯2 are again in good discriminating positions, each of them is positioned in a locally optimal position. In addition, the duplication of one of these hiddens, which are already in optimal discriminating positions, also results in a node being an optimal discriminator. Therefore, the system once again ends up in a false local minimum.

2.3 Type 3 : Inactive hidden nodes

In this case *all* the hidden nodes tend to be inactive (producing approximately zero outputs) for the samples in a certain region of the feature space. The example of the type 1 false local minimum depicted in Figure 5 also shares this feature. However, in that case the departure of a hidden node from the sample region left the system with too few nodes to solve the problem, in which case the failure of the classifier can be ascribed to the straying of a node. When four hidden nodes are used to attempt solving the problem, the straying of one of these leaves the network with three hidden nodes, which is adequate to develop an efficient classifier with.

Figure 6 displays the decision boundaries of a four-hidden-network which ended up in a false local minimum of type 3. Examples of networks developing in good classifiers after the straying of a hidden node had occurred, have been witnessed. The stray node in Figure 6 can therefore not be the sole cause of the classifier's failure. The difference between the classifier which performed well in spite of the straying of one of its nodes and the one depicted in Figure 6, is that the latter developed an area where all the hidden nodes are inactive. (Shaded region in Figure 6).

The sigmoidal decision boundaries constructed by these hidden nodes are characterized by sharp transitions. This is caused by large absolute values of the hidden-to-output weights. This phenomenon can be explained as follows: The training process reaches a stage where changes in the values of the weights resulting in changes in the positions of the decision boundaries, do not produce a decrease in the value of the criterion function. Steep transitions usually result in fewer classification errors being committed when two adjacent clusters need to be separated. The absolute values of the hidden-to-output weights are therefore increased in an effort to reduce the error rate of the classifier.

A small change in the position of a decision boundary with a sharp transition can have a considerable effect on the value of the criterion function, since it can lead to the total misclassification of a correctly classified sample. Decision boundaries with sharp transitions therefore tend to be very 'rigid'. Since the inactive region of the sample space can only be eliminated by moving at least one decision boundary a large distance, there is no prospect of the training process escaping from this false local minimum.

2.4 Type 4 : 'The rest'

The three classes of local minima discussed thus far are all characterized by specific properties which are easy to discern. These are also the most commonly encountered types of local minima. However, there exist various other types of local minima which are not so clearly characterizable. We group these cases into the taxonomist's indispensable 'miscellaneous' category. As was the case with the other types, the members of this group are also characterized be an ineffective application of the available hidden nodes, e.g.: a few hidden nodes performing a task which could easily have been fulfilled by a single node. We now give an

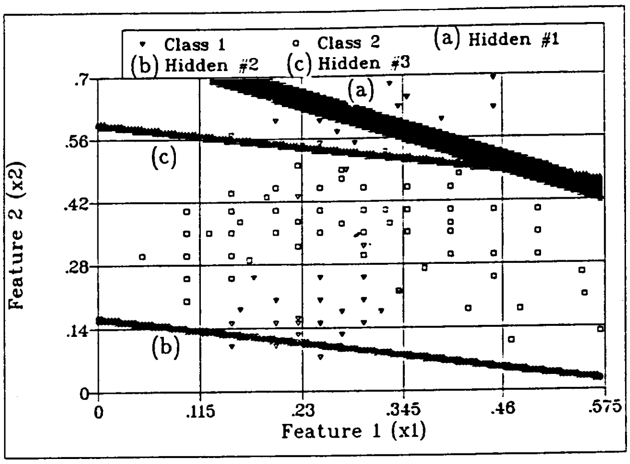


Figure 7: Decision boundaries of hidden nodes for a false local minimum of type 4

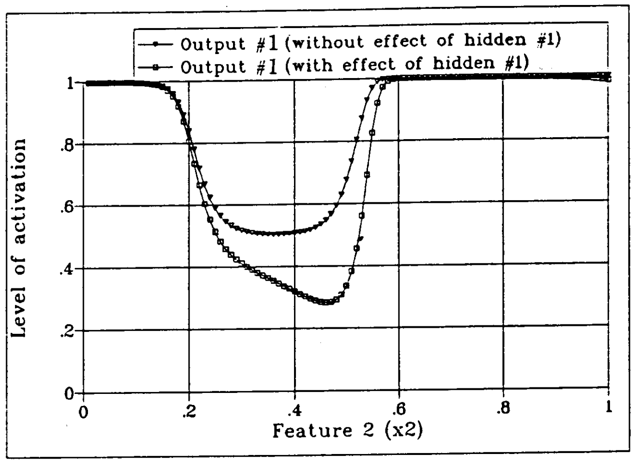


Figure 9: Variation of activity of output nodes along $x_1 = 0$

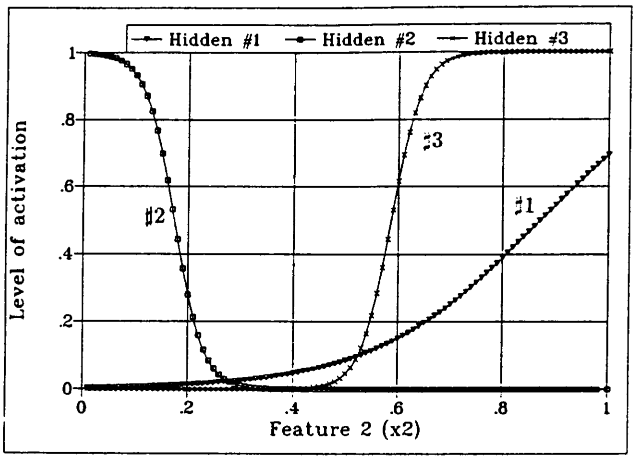


Figure 8: Variation in activity of hidden nodes along $x_1 = 0$

example of members of this group.

For the sake of explaining this example we need to examine the amplitude variations of the hidden and output nodes along the x_2 - axis (i.e. for $x_1 = 0$ in Figure 7). These amplitude variations of the hidden and output nodes are depicted in Figures 8 and 9 respectively. [3] A one-dimensional examination is adequate for this two-dimensional problem because the decision boundaries of the hiddens are more or less parallel to each other and approximately perpendicular to the x_2 axis. The plots in Figures 8 are therefore representative of all other one-dimensional plots of this nature which run parallel to the x_2 axis.

It can be seen from Figure 8 that hiddens ♯2 and ♯3 discriminate for class 1, since they are both mainly active in the regions of sample space occupied by the samples belonging to class 1 (see e.g. Figure 3) and since they are connected to the output nodes in such a fashion that the output representing class 2 (output ♯2) is inhibited and the output representing class 1 (output ♯1) is stimulated. Hidden ♯1, on the other hand, has a large positive weight leading to output ♯2 and a large negative connection to output ♯1. When considering these weights, it seems as if this hidden node is discriminating for class 2. Its active region, however, coincides with that of hidden ♯3 which is discriminating for class 1. Hidden ♯1 therefore seems to have the tendency to oppose hidden ♯3, since it neutralizes the effect of ♯3 on the outputs. This is the case to a certain extent; however, Figure 8 indicates that the slope of the sigmoid of ♯1 is much more gradual than that of ♯3. The contribution of ♯3 to the output nodes is therefore much bigger than that of ♯1, for $x_2 > 0.6$ (where the samples belonging to class 1 tend to occur). Hidden ♯3 therefore still succeeds in forcing the output nodes into the desired state, preventing an increase in the value of the criterion function due to the presence of hidden ♯1. In the region where the samples of class 2 are present, both hidden ♯2 and ♯3 are inactive. As can be seen from Figure 9 this would result – without taking ♯1 into account – in the activities of the output neurons both equalling 0.5. Since hidden ♯1 is more active than hidden ♯3 in this region, the consequence is a reduction in the value of the output for class 1 and an increase in the value of the output representing class 2. This is a better approximation to the desired vector for class 2, causing a reduction in the value of the criterion function. Moving the decision boundary of hidden ♯1 closer towards that of hidden ♯3 will clearly result in an increase of the criterion function since hiddden ♯1 will oppose ♯3 too strongly, causing the outputs to produce a vector of activities which is a worse approximation of the target vector of class 1. Moving the decision boundary of hidden ♯1 further away from hidden ♯3, will reduce its level of activity in the region occupied by class 2, leading to a cancellation of the reducing effect it had on the criterion function. Thus, moving hidden ♯1 either way results in an increase in the value of the

[3] The reader should bear in mind that Figure 7 only depicts the decision boundaries, and not the full variation in activation of the hidden nodes over the whole feature plane.

criterion function. Bearing in mind that the other hiddens developed into class discriminators which means that their positions are also locally optimal, the current state of the network therefore represents a local minimum.

Hidden ♯1 is, however, not utilized to its full potential. From Figure 4 it can be seen that its decision boundary has to move to the vicinity of that of hidden ♯2, in order to facilitate the curved decision boundary which separates the lower cluster of class 1 from classs 2. As already explained above, moving hidden ♯1 to such a position causes an increase in the value of the criterion function. Since the optimizer used for the training process is a *descent* technique, increases in the criterion function are impossible. Escaping from this minimum, which turns out to be a *false* local minimum, is therefore impossible.

3 Conclusion

By studying a specific example, we have found that there exist various classes of causes for the occurrence of false local minima. This understanding suggests methods of decreasing the probability of falling into such minima - for example, stray hidden neurons can be avoided to some extent by ensuring that all hidden neurons initially have their transition regions in regions of feature space populated by samples. We hope to be able to report on the success of such methods in the near future.

References

[1] D E Rumelhart, G E Hinton and R J Williams, [1986], 'Learning internal representations by error propagation,' in *Parallel Distributed Processing.* Cambridge MA: MIT press, pp 318-362, ch. 8.
[2] R P Lippmann, [1987], 'An introduction to computing with neural nets', *IEEE ASSP Magazine*, Vol. 2, pp. 4 - 22.
[3] W Y Huang and R P Lippmann, [1988], 'Neural net and traditional classifiers', in *Neural Information Processing Systems*, D Z Anderson, editor, Denver, Co, pp. 387 - 396.
[4] P D Wasserman, [1989], *Neural Computing: Theory and Practice*, Van Nostrand Reinhold, New York.
[5] R O Duda and P E Hart, [1973], *Pattern classification and scene analysis*, Wiley, New York.
[6] R L Watrous, [1987], 'Learning algorithms for connectionist networks: applied gradient methods of non-linear optimization,' *International Conference on Neural Networks*, IEEE, San Diego, pp. II-619 - II-627.
[7] P E Gill and W Murray and M H Wright, [1981], *Practical optimization*, Academic Press, London.
[8] R Hecht-Nielsen, [1989], 'Theory of the backpropagation neural network', *International Joint Conference on Neural Networks*, Washington, DC, pp. I-593 - I-606, IEEE.
[9] E Barnard, Personal communication.
[10] A Lapedes and R Farber, [1988], 'How neural nets work', in *Neural Information Processing Systems*, D Z Anderson, editor, Denver, Co, pp. 442-456.

Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research papers. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as a Communications or Viewpoints. While English is the preferred language of the journal papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted **in triplicate** to the editor.

Form of Manuscript

Manuscripts for review should be prepared according to the following guidelines.

- Use double-space typing on one side only of A4 paper, and provide wide margins.
- The first page should include:
  - title (as brief as possible);
  - author's initials and surname;
  - author's affiliation and address;
  - an abstract of less than 200 words;
  - an appropriate keyword list;
  - a list of relevant Computing Review Categories.
- Tables and figures should be on separate sheets of A4 paper, and should be numbered and titled. Figures should be submitted as original line drawings, and not photocopies.
- Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin, if they are not clear in the text.
- References should be listed at the end of the text **in alphabetic order** of the (first) author's surname, and should be cited in the text in square brackets. References should thus take the following form:

  [1] E Ashcroft and Z Manna, [1972], The translation of 'GOTO' programs to 'WHILE' programs, *Proceedings of IFIP Congress 71*, North-Holland, Amsterdam, 250-255.

  [2] C Bohm and G Jacopini, [1966], Flow diagrams, Turing machines and languages with only two formation rules, *Comm. ACM*, **9**, 366-371.

  [3] S Ginsburg, [1966], *Mathematical theory of context free languages,* McGraw Hill, New York.

Manuscripts *accepted* for publication should comply with the above guidelines, and may be provided in one of the following formats:

- in a **typed form** (i.e. suitable for scanning);
- as an **ASCII file** on diskette; or
- as a **WordPerfect**, **TEX** or **LATEX** or file; or
- in **camera-ready** format.

A page specification is available on request from the editor, for authors wishing to provide camera-ready copies. A styles file is available from the editor for Wordperfect, **TEX** or **LATEX** documents.

Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect scanning, typesetting, reproduction and other costs. Currently, the minimum rate is R20-00 per final page for camera-ready contributions and the maximum is R100-00 per page for contributions in typed format.

These charges may be waived upon request of the author and at the discretion of the editor.

Proofs

Proofs of accepted papers will be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Note that, in the case of camera-ready submissions, it is the author's responsibility to ensure that such submissions are error-free. However, the editor may recommend minor typesetting changes to be made before publication.

Letters and Communications

Letters to the editor are welcomed. They should be signed, and should be limited to about 500 words.

Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertisement

Placement of advertisements at R1000-00 per full page per issue and R500-00 per half page per issue will be considered. These charges exclude specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.

South African Computer Journal

Number 5, September 1991
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 5, September 1991
ISSN 1015-7999

Contents