II Simpósio Brasileiro de Redes Neurais, pp. 167-172, São Carlos - SP, Brasil, 1995.

A Bootstrap-like Rejection Mechanism for Multilayer Perceptron Networks

Germano C. Vasconcelos[1]

Departamento de Informática
Universidade Federal de Pernambuco
CP 7851, CEP 50732-970, Recife-PE, Brasil
E-mail : gcv@di.ufpe.br

Michael C. Fairhurst
David L. Bisset

Electronic Engineering Laboratories
University of Kent at Canterbury
Canterbury, Kent CT2 7NT, U.K.

Abstract

Multilayer perceptron networks (MLP) are investigated with respect to the problem of the detection of spurious patterns. A novel mechanism is proposed based on the ideas of bootstrapping that when incorporated into the standard MLP provides the network with the ability to continuously modifying its responses across the input space. The mechanism makes use of the outputs given by the network itself during its recall phase to improve performance in the rejection of spurious inputs, through a scheme of reinforcement of the classification decisions taken by the network. A example of the operation of this mechanism is shown in the classification of handwritten characters.

[1]Neural Networks, Pattern Recognition and Image Processing.

1 Introduction

The detection of 'spurious' or 'novel' [5, 7] patterns is an issue of great practical importance to many applications of neural networks. For example, in a factory control process managed by a neural network it is important to identify spurious operating conditions as distinct from normal conditions since they may indicate a novel failure mode which has not been seen during training. One of the first works to investigate the spurious pattern problem in the case of the recognition of 'random' patterns in MLPs was carried out in [8]. In [5], a more complete examination of the problem was presented.

The *bootstrap* algorithm [1, 2] is a mechanism that provides a way of adapting the weights of a processing unit in a neural network when the desired output for each training pattern is not known in an application environment. This algorithm is one of the basic principles of the paradigm of reinforcement learning [2] and has been successfully applied in practical applications [3]. Here, it is shown how a *novel* adaptation mechanism based on the concept of bootstrapping can be incorporated into an MLP network to improve its knowledge of the pattern space in the context of classification problems. It is shown how the decision regions generated by the network pre-trained on a set of classes through supervised learning can be gradually modified, during its recall phase, to improve its ability to detect spurious patterns. An example of the operation of the mechanism is given in the classification of handwritten characters.

2 MLPs and the Rejection Problem

In classification problems using feedforward neural networks, there are basically two ways of tackling the problem of detecting spurious inputs [5, 6]. The first is through the introduction of an additional mechanism operating in parallel and integrated with the network, and the second is based on the built-in properties of the network itself where the network's outputs are the only existing information for validating a classification decision. This work investigates MLPs [4] in the latter case. The motivation for this study is the characteristic observed in the standard MLP trained with backpropagation of the classification, with a high degree of confidence, of patterns with random characteristics (spurious patterns) as if they were legitimate members of the trained classes [5, 6, 8].

3 A Bootstrap-like Rejection Mechanism

One of the first techniques proposed to overcome the spurious pattern problem in MLPs was based on the idea of training the network with 'negative' examples of random patterns [8]. The objective is to create attractors in the pattern space representing the 'rejection' class so that when a pattern different from the training classes is presented to the network it is expected to be assigned to the rejection class rather than to any of the training classes. Unfortunately, this method does not provide a reliable solution to the problem [5, 6] since it is very much dependent on the unrealistic expectation that the random patterns arbitrarily chosen to represent unseen patterns correspond to good representatives of the portions of the input space desirable to be considered as rejection areas. However, the original concept of using negative training can be further considered in the recall phase of the network. Since, in the network's training phase, there is little (or no) a priori information about the kinds of patterns that should be rejected in a certain environment, one possibility is to use the classification decision taken by the network itself to dynamically change its knowledge about the structure of the pattern space.

A *bootstrap-like* rejection procedure can be described as follows : when a pattern is rejected by the network as a result of "great uncertainty" about its identity, it is "assumed" that the decision taken by the network was in fact correct and the network is re-trained to reinforce its decision. The responses of the output units are evaluated to determine if the output of the winning unit exceeds those of the others by more than a *confidence level*. If so, the input pattern presented to the network is accepted as belonging to the corresponding trained class. Otherwise, the pattern is rejected. In the latter case, a bootstrap control mechanism is responsible for checking the degree of "uncertainty" present in the rejection response of the network. In a classification decision, a rejection decision occurs either when the responses of all the output units are very low (close to '0') or some (or most) of the outputs are high (close to '1'). In either case, if the rejection is taken because the difference between the largest output and the second largest output did not reach a minimum defined *certainty level* then it is assumed that the pattern is very different from the training classes and can be used to re-train the network negatively to reinforce its rejection (with the vector of desired outputs defined as [0,0,. . .,0]). The network enters a process of relearning making use of a new on-line created training set composed of the rejected pattern and a sample of other patterns from the original training set.

The idea behind this approach is that other patterns similar to the current rejected pattern, which were still being erroneously classified, also come to be rejected as a result of the modification in the decision regions generated by the network. The mechanism of reinforcement described here is different from that of the classical paradigm of reinforcement learning since there is no "critic" involved in determining the desired output of the network based on the successes or failures of the system [2], and in the sense that the actual output of a processing unit is not thresholded at zero to determine the desired output for that unit [2]. Instead, the desired output of the entire network is determined through what can be called a transition from a state of "doubt" about the identity of the input pattern to a state of "certainty" that the input pattern is an spurious pattern and should, therefore, be rejected.

3.1 Visualization of the Decision Regions

A visualization procedure in a 2-class problem in a 2-dimensional input space can be used to demonstrate the inability of the MLP to deal with the rejection problem and to illustrate how the mechanism of bootstrapping can gradually modify the decision surfaces generated by the network. The experiment consists of training an MLP through backpropagation with a sample of patterns, and of testing the network response for each point in a large input matrix. The responses of the output units are compared and a pattern is classified as belonging to one of the 2 classes if the output (in the range [0,1]) for that class exceeds that of the other by a chosen confidence level (here taken as 0.65); otherwise, the pattern is rejected.

Figure 1(a) illustrates the result of the experiment where the elevated surfaces correspond to the points accepted by the network while the rest of the points represent the rejected patterns. It is seen that the vast majority of the points, even those most distant to the training patterns, are still accepted by the MLP. Only the points falling in between the training data points are rejected. The reasons for this are attributed to the use of the sigmoid as the network's activation function and to the fact that the backpropagation algorithm only places hyperplanes in between the training classes rather than around them, as would be more appropriate for the purpose of spurious pattern rejection [5, 6].

Figures 1(b),1(c) and 1(d) show the gradual effect of applying the bootstrap rejection mechanism. These show the modification of the decision regions after 1, 3 and 5 test epochs of presentation, respectively, of the whole matrix of points. The weights of the network are modified towards a more reliable structure for detecting spurious patterns through the creation of boundaries surrounding the training data; and, more importantly, these modifications happen without the action of an external teacher.

3.2 A Practical Pattern Recognition Application

Experiments have been carried out in a practical application. The aim is to train the network to recognize handwritten *digits* and to test its rejection performance with respect to spurious patterns, using for this purpose handwritten *alphabetic* letters. The database of characters is extracted from postcode information on envelopes in the UK mail to fit a 16 by 24 input matrix. The choice of this particular database was made with the intent of carrying out a rigorous test of rejection performance since digits and letters share many similarities. The set of digits is divided into two subsets of 1000 and 2000 characters, used respectively to train the network and to test its recognition performance. The letters consists of 2600 patterns with 100 patterns per class. MLPs composed of 80 units are run several times and the results are measured for different confidence levels imposed at the network's output for acceptance of a classification decision (from 0.15 to 0.65).

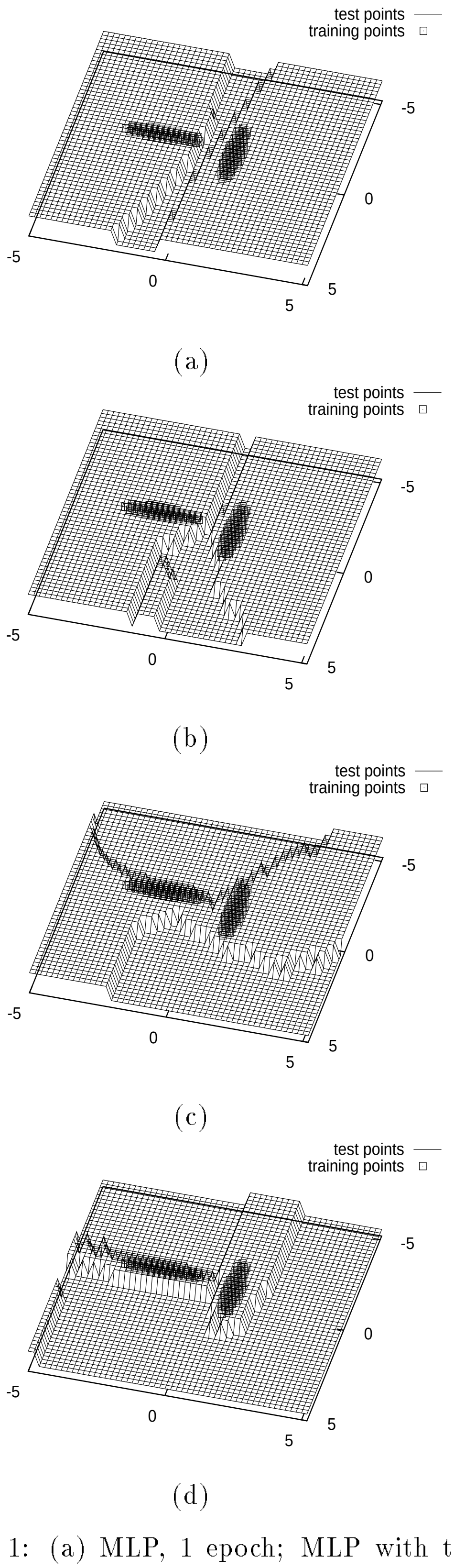
The first part of the experiments consists of presenting the test set of digits and letters to the network and of measuring the recognition and rejection rates both when there is no bootstrapping mechanism operating and when the mechanism is present. Furthermore, the results can illustrate the enhancement in performance that can be obtained with a test set of completely novel patterns, never seen before by the network. The second part consists of, again, presenting the database to the network but this time alternating the set of digits and letters several times to measure the evolution of the classification rates in the presence of bootstrapping. Although this second procedure cannot be considered completely realistic in a real environment, due to the fact that there is no repetition of patterns during the recall phase, it provides a way of qualitatively evaluating any possible change in performance, particularly in terms of degradation in the recognition rate for valid patterns.

The results obtained (Table 1) may first be compared in the absence of the bootstrap (epoch 0 in Table 1) with those obtained when the mechanism is operating (epoch 1 in Table 1). With the application of this technique the rejection rates for the letter test set are substantially increased. For a confidence level of 0.15, the rejection rate changes from 17 % to 32 % (an increase by a margin of 15 points over the original rate). For the other values of the confidence level, there is an improvement by 9.5 % on average. With respect to the correct recognition of digits there is no degradation in performance.

Letter Rejection						
	Confidence Level					
epochs	.15	.25	.35	.45	.55	.65
0	17	25	32	46	48	64
1	32	39	45	53	56	67
$\Delta 1 =$	+15	+14	+13	+7	+8	+3
3	53	52	61	63	66	67
6	61	64	67	70	73	76
9	66	68	70	74	76	76
12	68	69	71	76	77	79
15	68	70	72	75	78	79
$\Delta 15 =$	+51	+45	+40	+29	+30	+15

Digit Recognition						
	Confidence Level					
epoc.	.15	.25	.35	.45	.55	.65
0	88.1	88.1	84.4	80.8	81.3	76.0
1	89.4	88.3	85.9	84.5	82.0	81.0
$\Delta 1 =$	+1.3	+0.2	+1.5	+3.7	+0.7	+5
3	88.5	87.1	86.6	84.2	82.6	81.3
6	87.1	84.7	85.2	82.4	81.3	77.6
9	85.8	82.8	82.9	81.7	78.5	78.3
12	84.2	82.1	81.8	79.9	77.9	76.0
15	84.2	82.2	81.0	79.3	77.1	75.7
$\Delta 15 =$	-3.9	-5.9	-3.4	-1.5	-4.2	-0.3

Table 1: Letter rejection and digit recognition rates with the bootstrap mechanism



(a)

(b)

(c)

(d)

Figure 1: (a) MLP, 1 epoch; MLP with the bootstrap (b) 1 epoch, (c) 3 epochs, (d) 5 epochs.

In the second set of experiments, the network is run through a further 13 test epochs in the recall phase (represented by the epochs 3 to 15 in Table 1). It is possible to observe a gradual increase in the letter rejection rate with a further increase to 68 % after the 15^{th} epoch, for the value of 0.15 in the confidence level. With the other values an improvement by over 40 % is obtained after the 15^{th} epoch. Results measured in terms of the digit recognition rates show an advantageous trade off between reliability in rejecting invalid characters and some degradation in the recognition of valid patterns. It can be observed that initially there is no degradation in the digit recognition rate but, eventually, as the network runs through the test phase, the onset of degradation is observed. For the value of 0.15 for the confidence level after the 15^{th} epoch the decrease is by 3.9 % and for the other values it is by about 2.9 % on average. Another important characteristic is the fact that there is a substantial decrease in the digit error rates in all the cases reported as the network changes towards the more reliable structure. This is consistent with the objective of increasing the reliability of systems, where patterns for which there is low confidence about their identity are rejected, instead of being misclassified.

3.3 Combining Different Networks with Bootstrapping

The mechanism of bootstrapping can be integrated with other networks, more reliable than the MLP regarding the rejection problem, to further enhance their rejection capabilities. In [6], it was shown how the MLP can be turned into two more reliable structures through modifications in the network (the GMLP and the DMLP). The set of experiments carried out with the MLP when repeated with the GMLP and the DMLP shows a further increase in the rejection capabilities of these networks. For the GMLP, the rejection rate increases from 34 % to 47 % after the 1^{st} test epoch. This is an improvement of about 30 % over the performance of the MLP (17 %). After the 15^{th} pass 77 % of the letters are rejected by the GMLP. In the case of the DMLP, after the 1^{st} epoch, the increase in letter rejection is from 29 % to 41 %. After the the 15^{th} epoch, the letter rejection rate is improved further to 67 %. Once again, some trade-off is observed between establishing a more reliable structure and losing some performance in the correct classification ability. (See [5] for the full results obtained with the GMLP and the DMLP).

4 Final Remarks

A mechanism based on the ideas of bootstrapping was described to gradually modify the response of an MLP network across the input space, in the recall phase, in order to continuously improve its performance in the detection of spurious patterns. Although this mechanism is dependent on the occurrence of patterns rejected by the network originally for readaptation to take place, it provides a way of changing the network's knowledge of the pattern space without the need for supervised training. Moreover, it can be used with networks which are inherently more reliable than the MLP in this respect (the GMLP and the DMLP) to further enhance their rejection capabilities. In practical situations, the occurrence of spurious patterns should be much less frequent than that assumed in the experiments of letter rejection reported with the bootstrap mechanism. However, the experiments have been successful in demonstrating how the network performance can be altered to boost rejection performance in an interesting and powerfull unsupervised manner.

Future work will consider the actual implementation of the system described to solve a practical problem. A typical situation where this mechanism may be applied considers the control of a process in a factory in an environment characterised by the possible occurrence of abnormal situations. One strategy is to define an experimental phase where the system controls the process under constrained operating conditions until it reaches a point where it can be left operating by itself. In a real environment, some strategies for monitoring the performance of the system in its usage phase have to be considered in order to avoid any substantial degradation in the recognition of valid patterns. One of the possibilities is the estimation of, from time to time, the current system performance and its comparison with its original performance previously estimated on an independent test set of valid patterns (defined in a pre-usage phase) [5].

References

[1] Lucky, R.W. (1966). 'Techniques for adaptive equalization of digital communications systems'. Bell Syst. Tech. J. 45, 255-286.

[2] Widrow, B., Narendra, G.K., Maitra, S. (1973). 'Punish/Reward: Learning with a critic in adaptive threshold systems'. IEEE trans. on Syst., Man and Cyber., SMC-3, No.5, 455-465.

[3] Widrow, B., Stearns, S.D. (1985). 'Adaptive signal processing'. Prentice-Hall, Englewood Cliffs, NJ.

[4] Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). 'Learning internal representations by error propagation'. In Rumelhart, D.E., and McClelland, J.L.(Eds):'Parallel dist. proc., Vol. 1',, pp. 318-362. (MIT Press).

[5] Vasconcelos, G.C. (1995). 'An investigation of feedforward neural networks with respect to the detection of spurious patterns'. PhD Thesis, Electronic Engineering Labs, University of Kent, U.K.

[6] Vasconcelos, G.C., Fairhurst, M.C., and Bisset, D.L. (1995). 'Efficient detection of spurious inputs to improve the robustness of MLP networks in practical applications'. Neural Computing & Applic., Spring-Verlag, to appear.

[7] Bishop, C. (1994). 'Novelty detection and neural network validation'. IEE Proc. Vision, Image and Sig. Processing, Vol. 141, No. 4, 217-222

[8] Linden, A., Kindermann, J. (1989). *Inversion of multilayer nets*. IJCNN Washington.