Appl. Math. and Comp. Sci., 1995, vol.5, No.4, 743–749

ON THE ROBUSTNESS OF LEARNING IN THE MULTI-LAYER PERCEPTRON

PAUL WILLIAMS*, ANDREW W.G. DULLER*

The back-propagation algorithm, for training multi-layer perceptrons tends to be slow to converge to a final solution and many methods have been proposed for improving this. One technique takes advantage of an alternative training error criterion, however, we show that this reduces the robustness of the learning in the presence of outliers in the input data. Two examples are used to show the characteristics of the learning methods, one a test problem and the other from a "real-world" problem.

1. Introduction

In this work we investigate the robustness of the back-propagation algorithm applied to multi-layer perceptrons (MLP) during the training phase under two different output error criteria; the standard mean square error criterion and an alternative logarithmic criterion (van Ooyen and Nienhuis, 1992). Van Ooyen employed this measure as an acceleration mechanism which removes the derivative function from the output layer of the neural network, thus increasing the speed of escape of the networks neurons from saturation early on in training. These saturation effects create lengthy plateaus in the standard mean square error curve.

However, Huang and Lippmann (1987) have shown that the derivative provides inherent robustness during the learning phase of back-propagation. An investigation into the effects of the use of the logarithmic measure on the robustness of learning was therefore undertaken.

2. Motivation

The presence of outliers in training data causes a problem in the training of neural networks and can be tackled by using one of three methods; discard the data containing outliers, by removing outliers from the data using a suitable data cleaning algorithm, or by relying on the in-built robustness of a suitable classification technique.

Discarding data must be considered a last resort since in most applications sufficient training data is hard to obtain. The removal of outliers from the data generally involves a large amount of computation and also requires the production of a suitable parametric model for the generation of the data. In many cases a steady state probability distribution model is not possible since the presence of a trend may constitute an important feature. The above alternatives both have major drawbacks and therefore the robustness of the back-propagation algorithm is an important feature in the efficient training of MLP's with data containing outliers.

* School of Electronic Engineering and Computer Systems, University of Wales, Bangor, e-mail: andy@sees.bangor.ac.uk, paulw@sees.bangor.ac.uk

3. Increasing Training Speed

The van Ooyen convergence acceleration method (1992) is designed to take advantage of the learning error criterion given by

$$E_1 = -\frac{1}{CN}\sum_{i=1}^{C}\sum_{j=1}^{N}\Big(t_{ij}\ln z_{ij} + (1-t_{ij})\ln(1-z_{ij})\Big) \tag{1}$$

instead of the standard mean square error criterion which is given by

$$E_2 = \frac{1}{CN}\sum_{i=1}^{C}\sum_{j=1}^{N}(t_{ij}-z_{ij})^2 \tag{2}$$

where C is the number of output neurons (classes), N is the number of input samples, z_{ij} is the actual output of the neural network at node i due to input sample j, and t_{ij} is the target value associated with the actual output z_{ij}.

Let w_{ip} be the value of the weight connecting the p-th previous hidden layer node to the i-th output node and y_p is the output of the previous layer neuron p. Analysis of the two error criteria leads to the following:

$$\frac{\partial E_2}{\partial w_{ip}} = (z_i - t_i)z_i(1-z_i)y_p \tag{3}$$

for eqn. (2) and

$$\frac{\partial E_1}{\partial w_{ip}} = (z_i - t_i)y_p \tag{4}$$

for eqn. (1) in which we can see the missing term is $z_i(1-z_i)$ which accounts for the derivative of the sigmoid function in the output neurons. The derivative of the sigmoid is thus removed from the error back-propagation equations owing to the use of the error criterion of eqn. (1).

The advantage of this is that the networks output neurons spend less time in their saturated regions, a cause of the plateau at the beginning and end of the output learning error curve.

4. Robustness in Learning

Huang and Lippmann (1987) show that the performance of the MLP exceeds that of the Bayesian classifier when the input data contains outliers. MLPs are considered to be robust during training since they place a window on the input data of each layer when the algorithm multiplies the input data by the bell shaped derivative of the sigmoid function. Only inputs that are close to the currently estimated decision boundary may alter the boundary to any degree. Inputs which are further away from this boundary have little effect on the movement of the decision hyper-plane. It is therefore expected that the standard MLP is robust to a small number of outliers.

The problem with using eqn. (1) is that the removal of the output layer derivative to improve convergence rate has the detrimental effect of reducing the inherent robustness of the output layer of the multi-layer perceptron, as can be seen from the results in the next section.

5. Experiments and Results

Two investigations have been undertaken, of neural network training performance on data which is known to contain outliers using both the standard MLP error criterion of eqn. (2) and that given by eqn. (1). The first experiment used a simple training problem (Huang and Lippmann, 1987) and the second was for an application from the electricity industry involving "real-world" data used in analysing the "flicker" on domestic lighting circuits.

5.1. Test Problem

Consider the uniformly distributed data shown in Fig. 1 which shows two classes "A" and "B" of data, together with a set of outliers, A', which consists of 1% of the data contained in class "A". Class "A" has 990 uniformly distributed values between 0.0 and 0.1, class "B" has 1000 uniformly distributed values between 0.25 and 0.75, and the outlier "class", A', contains 10 values between 0.9 and 1.

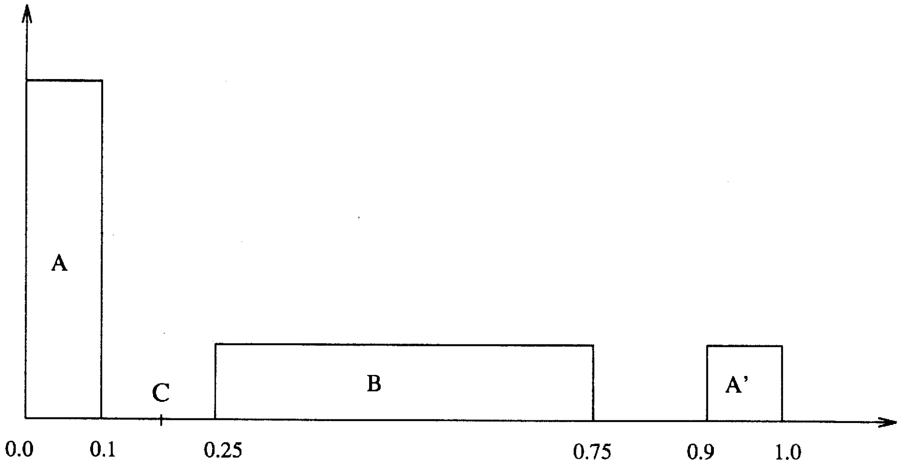

Fig. 1. A uniformly distributed data set in two classes, with outliers.

The principle of the experiment was to show that if a network is robust then the outliers of class "A" will be ignored (and placed in the other class). This problem was tackled using a 2-layer MLP with structure 1-4-2 (this is the same structure as that investigated by Huang and Lippmann (1987)), with a learning rate of 0.25 and a momentum factor of 0.7. In one case van Ooyen convergence acceleration was used and in the other the standard back-propagation algorithm was used; and the initial weights for all networks investigated in this paper were set randomly in the range $\langle -0.03, 0.03 \rangle$. Since the difference in the number of data samples available range $\langle -0.03, 0.03 \rangle$. Since the difference in the number of data samples available in "A" and "B" is only very small, the generalisation results may be considered to be independent of this small variation. The standard MLP ignored the 10 outliers as belonging to class A; and since the input must be classified in one of the classes (Richard and Lippmann, 1991), the network elected to classify the outliers as class B. The van Ooyen accelerated network classified the outliers as class A rather than ignoring their class designation; it can therefore be said that the criterion of eqn. (1) is less robust to outliers during learning than the standard back-propagation learning procedure using eqn. (2).

In all cases of classification the output node activations were > 0.9 for class membership and < 0.1 for class the input sample not belonging to that particular class; the neural networks may therefore be considered to be very confident in their classification decisions.

An important factor to consider when assessing the robustness of a network is the networks size (number of hidden neurons). A large network has more degrees of freedom with which to learn a given problem, whereas a smaller network is much more rigid during the learning process. This effect is referred to as network flexibility; a low flexibility (small) network generally will out perform a high flexibility (large) network during the generalisation process.

It is also the case that for some simple problems using *very* small networks robustness to outliers may be high, whether accelerated convergence or standard back-propagation is used. For example, a network with structure $1-1-2$ is just as robust to outliers in the accelerated version as in the standard version since there is only a single hyperplane available for hidden layer decision making. Owing to the large probability density of training samples around the point C in Fig. 1, following training the network will place the single hyperplane between classes A and B automatically classifying the outliers, A', into class B. In addition, a $1-2-2$ network was investigated using this simple problem and it was found that *both* available hyperplanes separated classes A and B. The exact reason for this is unknown, however it may be due to very similar initial weights for both hidden neurons and thus the splitting of the hyperplanes does not take place; this splitting normally takes place later on in learning when the neurons leave the linear region around the centre of the sigmoidal activation function.

5.2. Application to a Real World Problem

In order to illustrate more readily the effect of the accelerated convergence algorithm, a practical application will be demonstrated. The background to this application is that electricity companies receive numerous complaints from customers when the "flicker" on domestic lighting circuits reaches a certain threshold. This problem is generally caused by large pieces of electrical machinery producing voltage fluctuations in the supply. Each type of equipment causes a slightly different type of fluctuation which can be identified using a neural network (Williams, 1994; Williams and Duller, 1992). The training data for this application is often collected over a period of many hours (typically an eight hour shift) and this will be marked as being characteristic of the specific piece of equipment. However, this data may well contain samples taken when the machinery was switched off (e.g. a change of batch on a wire mesh welder plant) and these will produce outliers in the training set for a particular training data class. The main goal of this section is to show that in applications where only few outliers are present, and the network size is suitably chosen, the outliers can be ignored (thus saving pre-processing time) and training may continue with the full data set.

Note that what may be outliers of one class may *not* be outliers of another class, in fact they *may* impinge in any other class feature space; so like the simple experiment above, it is expected that outliers will be classified in some other class.

5.2.1. Background to the Problem

The training data consists of 5 classes; arc-furnace, spot-welder, air-conditioning equipment, large industrial motors (like those found in rock crushing plant) and steel rolling-mills. Over a measuring period of 10 minutes, the voltage levels owing to a particular disturbance are classified into 200 different "quantisation" voltage levels, a counter, associated with each quantisation level, is incremented each time the voltage appears in that level and at the end of the 10 minute period (for a sufficiently fast sampling rate) a probability distribution function is obtained. Figure 2 illustrates some typical 200 dimensional probability distribution functions of voltage fluctuations obtained as training data samples, together with a class called "background" which is effectively what remains on the supply when a typical disturbing load is disconnected from the network or switched off. These "background" samples are considered to be outliers in the flicker data.

5.2.2. Flicker Experiment

The training set consisted of 1000 samples of training data (200 samples from each of the 5 classes) plus a varying percentage (1% to 20%) of outlier samples. Owing to the distribution of the training classes in feature space it is possible to use a low number of hidden neurons (Williams, 1994). A $200 - H - 5$ network was used for the investigation into robustness (where H was varied from 3 to 5 neurons); these are relatively small networks, therefore low flexibility and high inherent robustness of the (un-accelerated) network can be assumed. In all cases the standard back-propagation algorithm displayed higher robustness than the accelerated version, although the degree of robustness depended on the percentage of outliers present and the number of hidden neurons as can be seen in the *Standard* column of Table 1 . In interpreting the results of the various training runs made on the flicker data, it was found that an extension to the original definition of robustness was required. This leads to the designations "type 1" and "type 2" for robustness. For the rest of this paper "type 1" robustness is said to occur when:

- an input outlier is given a class designation which is different to the one allocated to it during supervised training; the classification being determined by the highest output activation
- network output activations sum to 1 and so can be interpreted as probabilities.

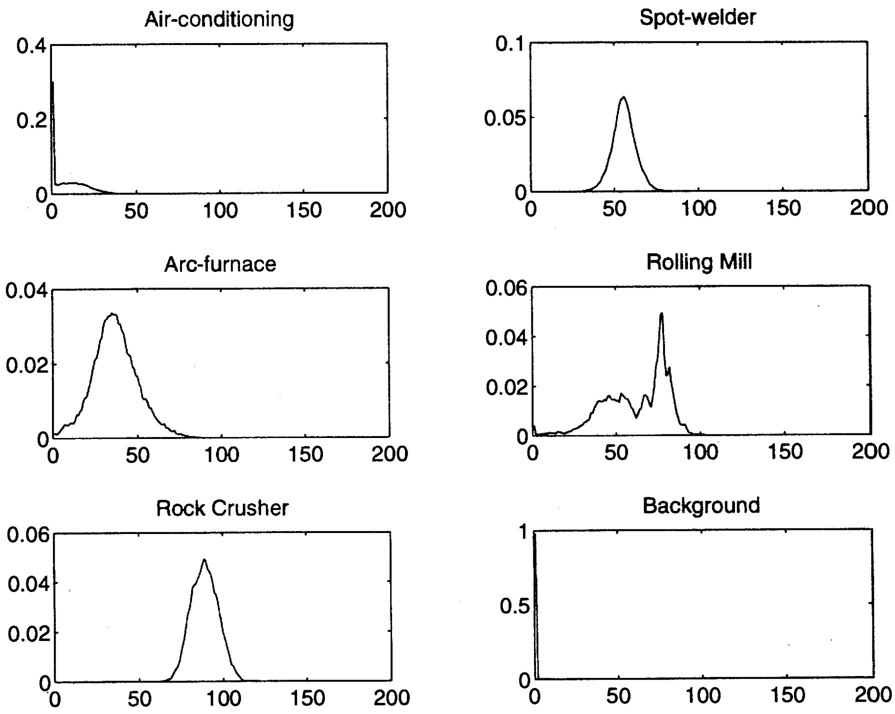

Fig. 2. Five classes of voltage disturbing equipment and outliers (background).

Tab. 1. Robustness results for standard back-propagation and van Ooyen enhanced back-propagation.

% Outliers	Hidden nodes	Standard robust?	van Ooyen robust?
1	3	yes (type 1)	no
3	3	yes (type 1)	no
5	3	yes (type 1)	no
7	3	yes (type 1)	no
10	3	yes (type 2)	no
20	3	yes (type 2)	no
1	4	yes (type 1)	no
3	4	yes (type 1)	no
5	4	yes (type 1)	no
7	4	yes (type 2)	no
10	4	yes (type 2)	no
20	4	yes (type 2)	no
1	5	yes (type 1)	no
3	5	yes (type 1)	no
5	5	yes (type 2)	no
7	5	yes (type 2)	no
10	5	yes (type 2)	no
20	5	no	no

"Type 2" robustness is said to occur when:

- an input outlier produces network output activations that do not sum to 1 and thus the outputs cannot be trusted; no matter what the output activation values are.

The table shows which network sizes were robust for each algorithm and a given percentage of one class being outliers (10% outliers from a class of 200 samples means 20 outliers were present in the data). For standard back-propagation, as the network size was increased the type of robustness was found to change from "type 1" to "type 2" and then eventually to fail. In general, the larger the network the less robust was the learning; also, the increasing percentage of outliers in the data reduced the apparent robustness of the network since the high probability of outliers in the data gave the impression of the outliers being valid training data. This high concentration of outliers lead the network to try harder to learn the outliers rather than effectively fully ignoring them. It can be seen that for all training runs made with the van Ooyen acceleration algorithm the network was never robust and always learned the outliers.

6. Conclusions

It has been demonstrated that for two layer MLPs the criterion given by eqn. (1) should not be used where the training data is suspected to contain outliers. This has been shown for both a test problem and a "real-world" application. In the case of the "flicker" problem the use of the original learning criteria led to a substantial increase in the robustness of the MLP over the van Ooyen method for various concentrations of data outliers and various network sizes.

Two types of robustness have been identified for standard back-propagation and it has been shown that network robustness degrades from "type 1" to "type 2" before robustness is lost as the percentage of outliers increases.

The degree of robustness of the MLP is likely to be problem dependent and an investigation of this is currently being performed with a number of other "real-world" applications.

References

Huang W.Y. and Lippmann R.P. (1987): *Comparisons between neural net and conventional classifiers.* — IEEE Int. Conf. *Neural Networks*, v.4, pp.485–493.

van Ooyen A. and Nienhuis B. (1992): *Improving the convergence of the back-propagation algorithm.* — Neural Networks, v.5, No.3, pp.465–471.

Richard M.D. and Lippmann R.P. (1991): *Neural network classifiers estimate Bayesian a posteriori probabilities.* — Neural Computation, v.3, No.4, pp.461–483.

Williams P. (1994): *An Investigation of Neural Networks for Fault Identification in the Electricity Supply Industry.* — PhD thesis, University of Wales, Bangor.

Williams P. and Duller A.W.G. (1993) *Identification of lighting flicker sources using a neural network*, In: Techniques and Applications of Neural Networks (M.J. Taylor and P.J.G. Lisboa, Eds.). — Hemel Hempstead, UK, Ellis Horwood, chapter 11, pp.183–197.

Received: May 30, 1995
Revised: October 16, 1995