Limiting the Effects of Weight Errors in Feed Forward Networks Using Interval Arithmetic

Davide Anguita, Sandro Ridella, Stefano Rovetta, and Rodolfo Zunino
University of Genova - D.I.B.E., Via Opera Pia 11A, 16145 Genova, Italy.
e-mail: {anguita,ridella,rovetta,zunino}@dibe.unige.it

ABSTRACT

We address in this work the problem of weight inaccuracies in digital and analog feed forward networks. Both kind of implementations suffer from this problem due to physical limits of the particular technology. This work presents a novel and effective approach through the application of interval arithmetic to the multi layer perceptron. Results show that our method allows to (1) compute strict bounds of the output error of the network, (2) find robust solutions respect to weight inaccuracies and (3) compute the minimum weight precision required to obtain the desired performance of the network

1. Introduction

As pointed out in [12], the training of neural devices is usually performed in three different ways: off-chip, chip-in-loop, and on-chip. The first technique would be preferable, when on-site learning is not required, because involves only the downloading of the weights computed once (and off-line) with a method of choice. Unfortunately, the weight inaccuracies, deriving from the physical implementation (e.g. limited numerical precision, circuit offsets, noise, etc.), can greatly reduce the quality and the usability of the programmed device.

The problem of weight inaccuracies in digital and analog implementations of multilayer perceptrons has been studied in the past. Several solutions have been proposed in the literature that deal with this problem, offering modified learning algorithms [5, 10] or statistical/heuristic techniques that analyse the effect of weight errors [3, 4, 8, 11, 13].

We propose here a new and effective method based on Interval Arithmetic [1]. In section 2 the Interval Arithmetic Multi Layer Perceptron (IAMLP) is presented that allows to bound the network output respect to weight errors. In the following section we present a learning algorithm that finds robust solutions and computes the minimum numerical precision required by the weights, given a learning problem and the desired error threshold.

2. Interval Arithmetic Multi Layer Perceptron

The Interval Arithmetic Multi Layer Perceptron (IAMLP) was introduced in [6] and has shown several interesting features [2, 7]. The rules of interval arithmetic allow the IAMLP to deal with real numbers as well as with intervals of the form $[w^L, w^U]$ with $w^L \leq w^U$ (note that real values are a particular case of intervals with $w^L = w^U$). We summarize here briefly the main equations of the IAMLP.

Let us consider two intervals $X = [x^L, x^U]$ and $Y = [y^L, y^U]$ with $y^L \geq 0$ for simplicity; we can define arithmetic operations $(+, -, \cdot)$ and the sigmoid function as follows:

$$X + Y = \left[x^L + y^L, x^U + y^U\right] \qquad X - Y = \left[x^L - y^U, x^U - y^L\right]$$

$$\mathrm{sgm}(X) = \left[\mathrm{sgm}\left(x^L\right), \mathrm{sgm}\left(x^U\right)\right] \quad X \cdot Y = \begin{cases} \left[x^L y^L, x^U y^U\right] & \text{if} \quad x^L \geq 0 \\ \left[x^L y^U, x^U y^U\right] & \text{if} \quad x^L \leq 0 \leq x^U \\ \left[x^L y^U, x^U y^L\right] & \text{if} \quad x^U \leq 0 \end{cases}$$

0-7803-3210-5/96 $4.00©1996 IEEE

Using the above rules, we can write the equations for a two-layer IAMLP in compact form:

$$O_i = \mathrm{sgm}\left(\sum_j W_{ij}^{(2)} H_j + W_{i0}^{(2)}\right) \qquad H_i = \mathrm{sgm}\left(\sum_j W_{ij}^{(1)} x_j + W_{i0}^{(1)}\right)$$

where $H_i, O_i, W_{ij}^{(l)}$ are intervals and x_j are real values. The intervals of the IAMLP can be used to model the inaccuracies of the network; in fact, each weight can be considered as composed by a fixed part plus an error: $W = [w^L, w^U] = [w - \varepsilon, w + \varepsilon]$ with $w = \frac{w^L + w^U}{2}$ and $\varepsilon = \frac{w^U - w^L}{2}$.

The intervals $O = [o^L, o^U]$ bound the outputs of the network and define precisely the behavior of the IAMLP when the weights are in the range defined by the above equation.

3. Bounding weight accuracies and finding robust solutions

A bp-like algorithm can be easily derived for the IAMLP if we define a suitable error function (see [7, 2] for details):

$$E_0 = \frac{1}{2}\sum\nolimits_{p,i} \left[\left(t_{p,i} - o_{p,i}^L\right)^2 + \left(t_{p,i} - o_{p,i}^U\right)^2\right] \tag{1}$$

Let us suppose that the weights of a network are distributed in the range $[-w_{max}, w_{max}]$ and let us quantize this range in $2N$ intervals of amplitude Δ. Obviously, to identify precisely one of these intervals we need $n_{bit} = \lceil \log_2(2N) \rceil$ bits.

Now, if we can guarantee the following condition:

$$\exists \quad -N \leq n < N : \quad [n\Delta, (n+1)\Delta] \subseteq \left[w_i^L, w_i^U\right] \forall i \tag{2}$$

we can safely use n_{bit} bits to represent each weight of the network. It is easy to show that (2) holds if we choose N large enough to cover the entire range $[-w_{max}, w_{max}]$ and

$$\Delta = \min_i \frac{(x_i^U - x_i^L)}{2} \tag{3}$$

If this is the case, we can write:

$$n_{bit} \geq \left\lceil \log_2 \left(\left\lceil \frac{4 w_{max}}{min_i \left(w_i^U - w_i^L\right)} \right\rceil - 1 \right) \right\rceil \tag{4}$$

(proofs are omitted due to space constraints). Eq. (4) gives the minimum precision (in bits) required to represent the weights with a total error: $E^* \leq E_0$. Note that we have assumed the same precision for all the weights; in some cases (e.g. digital implementations) this assumption could be relaxed and even less bits would be required.

A robust solution respect to weight errors can be obtained minimizing (4) or, in other words, maximizing the coarseness of the quantization. One way to obtain this is adding a penalty term to the error function (1):

$$E = E_0 - \lambda \sum_i \left(w_i^U - w_i^L\right) \tag{5}$$

This penalty term forces the intervals to grow, allowing for a coarser quantization of the weight range.

4. Experimental results

Let us consider a training set composed of 40 random points sampled in a unit square: the points lying inside a square of area $\frac{1}{2}$ and centered in $(0.5, 0.5)$ belong to the class C_1, the others to the class C_0 (Fig. 1a).

Fig. 1b (thick line) shows the discriminating boundary generated by the network ($output = \frac{1}{2}$) when the learning is stopped as soon as the patterns are correctly classified. The other six discriminating boundaries (thin lines) are obtained adding to the weights a small perturbation of $\pm 2^{-9} w_{max}$, $\pm 2^{-8} w_{max}$ and $\pm 2^{-7} w_{max}$ respectively. The effect on the network performance is obviously disastrous: even the smallest perturbation causes a misclassification of some patterns.

Fig. 1c shows the effect of stressing the learning to the limit ($E_0 \leq 10^{-4}$). The desired effect would be to push the discriminating boundaries farther from the border patterns. Yet, if we apply the same perturbations to the weights of this network, the number of misclassifications increases and the quality of the solution worsen. Furthermore, we cannot predict a-priori the performance decrease due the weight errors.

Fig. 1d shows a solution found by the IAMLP with $n_{bit} = 9$. The performance of the network is exactly as expected: the solution is robust to weight errors and there are no misclassifications if the weight errors fall inside the predicted range.

5. Conclusions

We have presented an application of the Interval Arithmetic Multi Layer Perceptron to the problem of weight errors due to network implementations. In the future, we plan to investigate the quality of the solutions found by this network, studying their relation to MDL principle [9].

References

[1] G. Alefeld and J. Herzberger, *Introduction to Interval Computation.* Academic Press, New York, 1983.

[2] D. Anguita, S. Ridella, S. Rovetta, and R. Zunino, "Incorporating a-priori knowledge into neural networks," *Electronics Letters*, Vol. 31, No. 22, pp. 1930–1931, October 1995.

[3] K. Asanović, N. Morgan, J. Wawrzynek, "Using simulations of reduced precision arithmetic to design a neuro-microprocessor," *Journal of VLSI Signal Processing*, vol. 6, no. 1, pp. 33–44, June 1993.

[4] J.Y. Choi and C.H. Choi, "Sensitivity Analysis of Multilayer Perceptrons with Differentiable Activation Functions," *IEEE Trans. on Neural Networks*, Vol. 3, No. 1, pp. 101–107, Jan. 1992.

[5] M. Höhfeld and S.E. Fahlman, "Probabilistic rounding in neural network learning with limited precision", *Neurocomputing* Vol. 4, pp. 291–299, 1992.

[6] H. Ishibuchi and H. Tanaka, "An Extension of the BP Algorithm to Interval Input Vectors: Learning from Numerical Data and Expert's Knowledge", *Int. Joint Conf. on Neural Networks*, Singapore, pp. 1588–1593, 1991.

[7] H. Ishibuchi, H. Tanaka, and H. Okada, "An architecture of neural networks with interval weights and its application to fuzzy regression analysis", *Fuzzy Sets and Systems*, Vol. 57, pp. 27–39, 1993.

[8] S. Piché, "The Selection of Weight Accuracies for Madalines", *IEEE Trans. on Neural Networks*, Vol.6, No.2, pp. 432–445, March 1995.

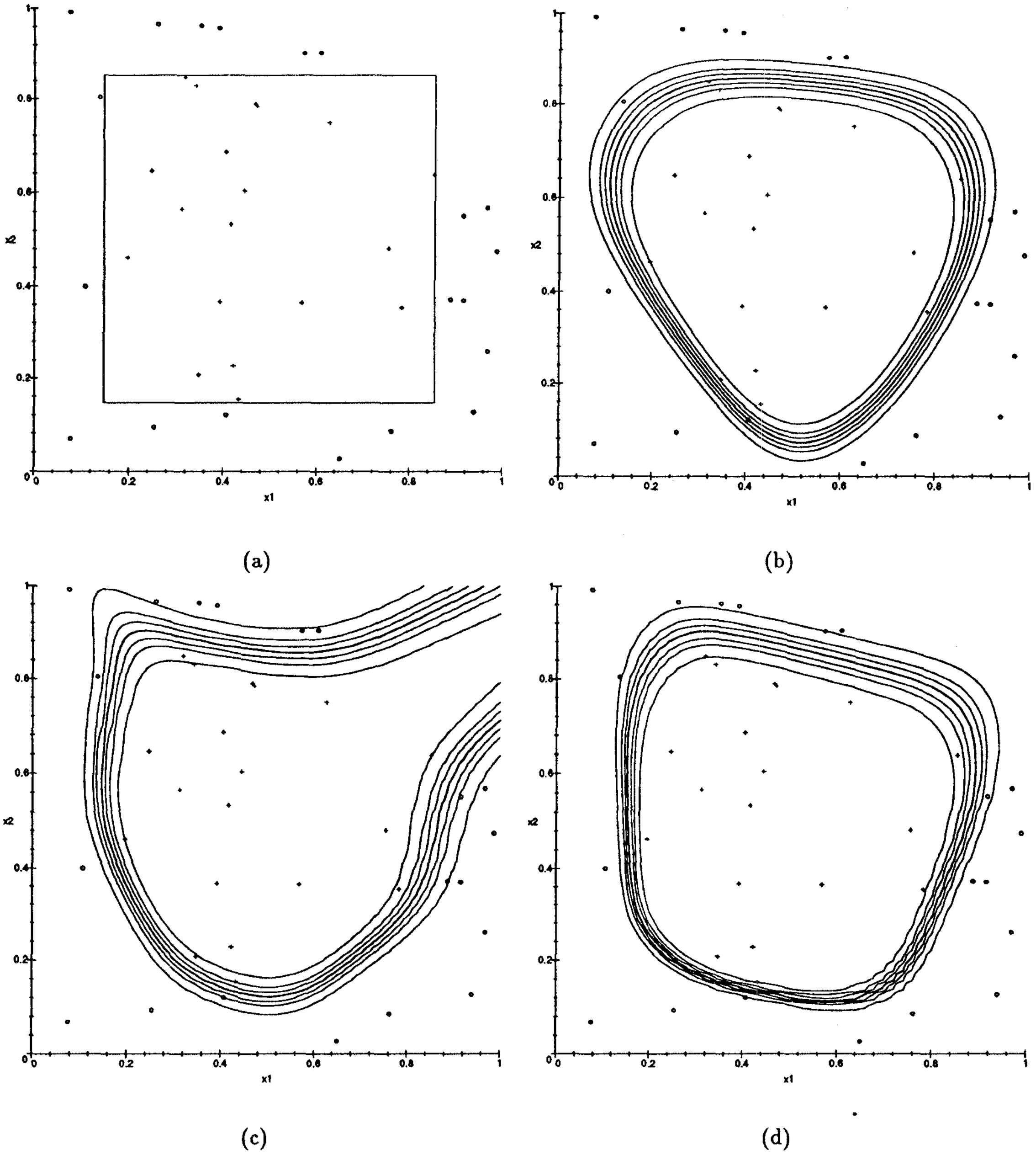
[9] J. Rissanen, *Stochastic Complexity in Statistical Inquiry.* World Scientific Pub., 1989.

[10] S. Sakaue, T. Kohda, H. Yamamoto, S. Maruno, and Y. Shimeki, "Reduction of Required Precision Bits for Back-Propagation Applied to Pattern Recognition", *IEEE Trans. on Neural Networks*, Vol. 4, No. 2, pp. 270–275, March 1993.

[11] M. Stevenson, R. Winter and B. Widrow, "Sensitivity of Feedforward Neural Networks to Weight Errors", *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, pp. 71–80, March 1990.

[12] G. Cairns and L. Tarassenko, "Precision Issues for Learning with Analog VLSI Multilayer Perceptrons", *IEEE Micro*, pp. 54–56, June 1995.

[13] Y. Xie and M.A. Jabri, "Analysis of the Effects of Quantization in Multilayer Neural Networks Using a Statistical Model", *IEEE Trans. on Neural Networks*, Vol. 3, No. 2, pp. 334–338, March 1992.



(a)

(b)

(c)

(d)

Figure 1: Effects of weight errors on the quality of learned solutions (see text for details)