Precision Requirements for Single-Layer Feed-Forward Neural Networks

A.J.Annema, K.Hoen and H.Wallinga

MESA Research Institute

University of Twente, P.O.Box 217,

7500 AE Enschede, The Netherlands

Abstract

This paper presents a mathematical analysis of the effect of limited precision analog hardware for weight adaptation to be used in on-chip learning feed-forward neural networks. Easy-to-read equations and simple worst-case estimations for the maximum tolerable imprecision are presented. As an application of the analysis, a worst-case estimation on the minimum size of the weight storage capacitors is presented.

1 Introduction

In neural networks, signal processing is in principle performed by simple processors (neurons) operating in parallel. Implementing neural networks in parallel hardware seems therefore natural. Because chip area is expensive, it is important to determine specifications for the various blocks composing the neurons, in order to reduce the chip area as much as possible without significantly degrading the neural network's performance. In this paper, the effect of limited precision analog weight adaptation circuitry on training is analyzed.

In the analyses, we use the so-called Vector Decomposition Method (VDM) which has been introduced recently [1],[2]. In section 2, a short introduction into the VDM and some resulting equations for single-layer feed-forward networks will be given. These equations will be used in the analysis of the effect on the learning behavior of limited precision weight adaptation blocks in section 3.

When implementing neural networks with learning capability in analog hardware, parasitic weight adaptation due to offsets, leakage and charge injection may occur. These effects can be modelled as an extra constant weight adaptation component for each adaptation cycle. In section 3, an analysis is given of the effect of constant weight adaptation on the learning behavior. At the end of section 3, a worst case estimation of the maximum tolerable constant weight adaptation is given.

Section 4 gives an application of the analysis and section 5 finally summarizes the conclusions.

2 Dynamics of single-layer networks: a summary

In this section, first a short summary of the essentials of the so-called vector decomposition method (VDM) [1],[2] are summarized. After this summary, the most important results of the analyses of single-layer feed-forward neural nets in [1] are presented.

The VDM is based on the introduction, for every neuron in a neural network, of a new base. The bases are used to decompose the weight vector and input vector of all neurons into three orthogonal vector components. The three vector components are respectively:

- related to only the bias input signal (denoted with a *bias* superscript),
- perpendicular to the attractor hyperplane (denoted by an h superscript),
- in parallel to this attractor hyperplane (denoted by either an F or an ε superscript).

The attractor hyperplane is defined as the hyperplane that results in a local (or momentary) optimal performance by the neural network. For single-layer neural networks, this attractor hyperplane is in general quasi stationary in input space. It appears that decomposing the weight vector and input vector of all neurons in vector components that are related to the attractor hyperplane of the neurons results in easy-to-read equations.

With the VDM, the weight vector $\underline{W}$ of the neuron is decomposed as

$$\underline{W} = \beta^h \underline{B}^h + \beta^{bias} \underline{B}^{bias} + \underline{W}^{\varepsilon} \qquad (1)$$

0-8186-6710-9/94 $4.00

and input vector $\underline{U}$ of the neuron is

$$\underline{U} = \alpha^h \underline{B}^h + \alpha^{bias} \underline{B}^{bias} + \underline{U}^F, \quad (2)$$

where the $\underline{B}^h$ and $\underline{B}^{bias}$ vectors are unit vectors.

The α^h and β^h may be associated with the concepts of *relevant information* and *correct knowledge* respectively. Similarly, $|\underline{U}^F|$ and $|\underline{W}^\varepsilon|$ may be associated with *irrelevant information* and *incorrect knowledge* respectively.

With the VDM, it has been derived [1] that the adaptation of the norm of the weight vector component in the local optimal direction (excluding the bias related weight) is given by

$$\Delta\beta^h \approx \frac{\eta'}{P} \sum_p \left(D^p - f(\underline{W}\cdot\underline{U}^p)\right) f'(\underline{W}\cdot\underline{U}^p)\, \alpha^{h,v,p} - \frac{\alpha^{ATT}\, \underline{U}^{F,const} \cdot \left(\Delta\underline{W}^\varepsilon\right)_{w\varepsilon}}{\alpha^{bias^2} + |\underline{U}^{F,const}|^2 + \alpha^{ATT^2}} \quad (3)$$

$$\text{with} \quad \eta' = \eta \left(\frac{\alpha^{bias^2} + |\underline{U}^{F,const}|^2}{\alpha^{bias^2} + |\underline{U}^{F,const}|^2 + \alpha^{ATT^2}} \right)$$

In these equations:

- p denotes the index of the example; the total number of examples is P
- α^{ATT} is the average part of α^h (for more precise definition see [1])
- $\alpha^{h,v}$ corresponds to the zero-mean part of α^h, with $\alpha^h = \alpha^{h,v} + \alpha^{ATT}$
- D is the desired (or target) response of the neuron for an example
- $\underline{U}^{F,const}$ is the average part of $\underline{U}^F$ (for more precise definition see [1])
- η is the adaptation factor

The first term on the right hand side of (3) corresponds to the ideal adaptation of β^h, i.e. corresponds to the ideal adaptation of the weight vector in the local optimal direction (excluding the bias related weight). The adaptations of the $\underline{W}^\varepsilon$ and of the β^{bias} of the weight vector will not be analyzed in this paper for reasons of compactness. The adaptations of the $\underline{W}^\varepsilon$ and of the β^{bias} are however used to derive (3).

3 Estimation of precision requirements

In analog on-chip learning feed-forward neural nets, analog circuitry takes care of the adaptation of every weight. Because the weights are usually stored as voltages across capacitors, the adaptation circuitry is typically an analog multiplier with charge output. This charge is usually constructed by gating an output current during some predefined interval (τ) to the weight storage capacitor C_{weight} [1],[3]-[8].

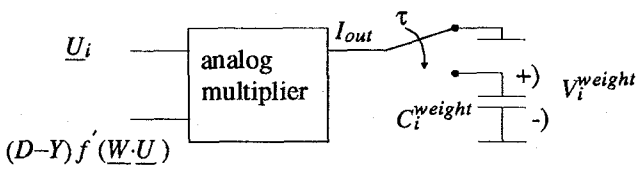

Fig. 1 Analog hardware realization of weight storage and adaptation circuitry

The weight adaptation is the sum of the ideally wanted weight adaptation and a non-ideal (or parasitic) weight adaptation:

$$\Delta\underline{W}_i^{total} = \Delta\underline{W}_i + \Delta\underline{W}_i^{par} \quad (4)$$

The parasitic weight adaptation is generally caused by offsets in the multipliers, offsets in its input signals, leakage of the stored weight-voltage, or is caused by charge injection of the switch that gates the output current of the multiplier to the capacitor [1],[9]. The parasitic weight adaptation due to offsets and due to charge injection is approximately proportional to the number of weight adaptations because these effects occur only during adaptation. The leakage effect is constant in time and therefore independent of the number of weight adaptations. However, it is assumed that the effect of leakage is small compared to the summed effects of the offsets and of the charge-injection. It follows that the parasitic weight adaptation is by approximation constant for each weight adaptation.

In the remainder of section 3, a worst-case estimation for the maximum value of the parasitic weight adaptation will be given. This estimated maximum corresponds to the value of the constant weight adaptation, $\Delta\underline{W}_i^{par}$, for which the eventual weight vector is close to the eventual optimal weight vector (i.e. the weight vector corresponding to the global or local minimum in the energy landscape). This estimation results therefore in specifications for analog weight adaptation blocks as will be shown in section 4.

3.1 Estimation of the MSE increase due to constant weight adaptation

In this subsection, an analysis of the effect of constant weight adaptation on the learning behavior of single-layer feed-forward networks is presented for a relatively simple case; it is assumed for simplicity reasons that the effect of the $\underline{U}^F$ vector components on training are negligibly small. In [1] this assumption lead towards the condition $|\underline{U}^{F,const}| = 0$. In a loose way , this condition means that the average (over the training examples) of any element of the $\underline{U}^F$ vector is zero. As a result of this, it can be derived [1] that the weight vector follows during learning a straight path in weight space from an initially point (assumed to be close to the origin of the weight space) towards the eventual spot. Because of this assumption, the second term on the right hand side of (3) is zero.

The parasitic adaptations of the weights form a parasitic adaptation vector $\Delta\underline{W}^{par}$, which is decomposed into three vector components related to the attractor hyperplane of the neuron:

$$\Delta\underline{W}^{par} = \beta^{h,par}\underline{B}^h + \beta^{bias,par}\underline{B}^{bias} + \underline{U}^{F,par} \qquad (5)$$

With the two equations that describe the adaptation of β^h and β^{bias}(comparable to (3)), it can be derived that the total adaptation of the β^h is given by

$$\Delta\beta^h \approx \frac{\alpha^{bias^2}}{\alpha^{bias^2}+\alpha^{ATT^2}}\left[\frac{\beta^{bias,par}\alpha^{ATT}}{\alpha^{bias}} - \beta^{h,par} + \right. \qquad (6)$$

$$\left. \frac{\eta}{P}\sum_p \left(D^p - f(\underline{W}\cdot\underline{U}^p)\right) f'(\underline{W}\cdot\underline{U}^p)\alpha^{h,v,p}\right]$$

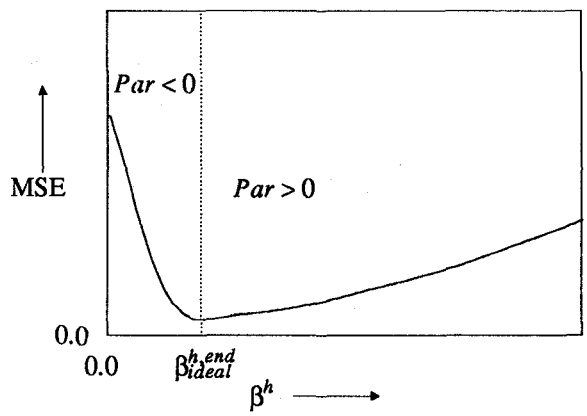

Fig. 4 Simulated MSE curve during training (assuming a small $\underline{W}^\varepsilon$); dotted line corresponds to the global minimum; $Par = (\beta^{bias,par}\alpha^{ATT})/\alpha^{bias} - \beta^{h,par}$

It follows from (6) that if the sum of the first two terms between brackets is positive, there is a positive net adaptation of β^h in the global minimum. Because the *MSE* is a function of β^h (see figure 2), the MSE versus time curve has a minimum during training for this case; after reaching the minimum, the MSE will increase during continued training.

In case that $\beta^{h,end} > \beta^{h,end}_{ideal}$, the eventual fraction of correctly classified examples is larger than in case of an ideal neural network, which can be explained using figure 3. Figure 3 shows a two-dimensional non-linearly separable training set (note that for illustration reasons, $|\underline{U}^{F,const}| \neq 0$). During training, the target response of examples out of $class_0$ is D_0 and the desired response for $class_1$ examples is D_1. The hyperplanes for which the response of the neuron is either D_0 or D_1 are marked by the dotted lines in parallel to the attractor hyperplane.

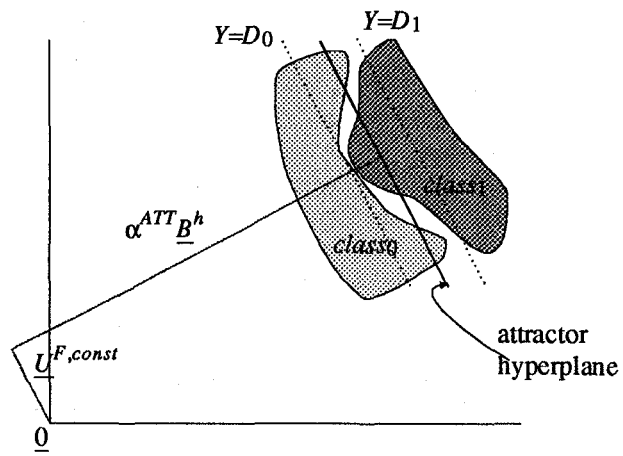

Fig. 3 Two dimensional training set in the input space; the two classes to be separated are denoted by $class_0$ and $class_1$

The distance between an example and the attractor hyperplanes is

$$\alpha^{h,v} = \frac{f^{-1}(Y)}{\beta^h}, \qquad (7)$$

and the distance between the hyperplanes for which $Y=D_0$ or $Y=D_1$ is

$$\Delta\alpha^h = \frac{f^{-1}(D_1) - f^{-1}(D_0)}{\beta_h}.$$

With this equation, it can now easily be seen from figure 3 that with increasing β^h, the fraction of correctly classified examples increases. Note that this is due to the fact that by increasing the β^h, only the fraction of correctly classified *linearly separable* examples increases whereas

the fraction of correctly classified *not-linearly separable* examples remains zero.

In case of a negative sum of the first two terms between brackets in (6), the neural network will not reach the global minimum because the eventual $\beta^{h,end} < \beta^{h,end}_{ideal}$.

The difference between the minimum attainable MSE for an ideal neural net and the MSE to be reached by the neural net with constant weight adaptation can now be approximated by

$$\Delta MSE \approx \frac{\beta^{h,end} - \beta^{h,end}_{ideal}}{2}\left[\frac{\partial MSE(\beta^h)}{\partial \beta^h}\bigg|_{\beta^{h,end}_{ideal}} + \frac{\partial MSE}{\partial \beta^h}\bigg|_{\beta^{h,end}}\right]$$

$$= \frac{\beta^{h,end} - \beta^{h,end}_{ideal}}{2} \cdot \frac{\partial MSE}{\partial \beta^h}\bigg|_{\beta^{h,end}}$$

where $\beta^{h,end}_{ideal}$ corresponds to the norm of the weight vector (excluding the bias-related weight) corresponding to the global minimum. Similarly, $\beta^{h,end}$ corresponds to the actual eventual value of β^h. The difference between $\beta^{h,end}$ and $\beta^{h,end}_{ideal}$ can be obtained from

$$\frac{\partial^2 MSE}{\partial \beta^{h^2}}\left(\beta^{h,end} - \beta^{h,end}_{ideal}\right) \approx$$

$$\frac{\partial MSE(\beta^h)}{\partial \beta^h}\bigg|_{\beta^{h,end}} - \frac{\partial MSE(\beta^h)}{\partial \beta^h}\bigg|_{\beta^{h,end}_{ideal}}.$$

Noting that the right hand side of (3) is identical to $-\eta$ times the first derivative of the *MSE* with respect to β^h [10], it follows directly that under the assumption of $\underline{U}^{F,const} = \underline{0}$

$$\Delta MSE \approx \frac{\left(\beta^{h,par} - \beta^{bias,par}\frac{\alpha^{ATT}}{\alpha^{bias}}\right)^2}{2\eta^2 \frac{\partial^2 MSE}{\partial \beta^{h^2}}} \qquad (8)$$

Using a Taylor series expansion, it is straight forward to show that

$$\frac{\partial^2 MSE}{\partial \beta^{h^2}} \approx \frac{-2}{1 - P(correct)} \cdot$$

$$\int \mathrm{P}(D-Y)\left[(D-Y)f''(\underline{W}\cdot\underline{U}) - f'^2(\underline{W}\cdot\underline{U})\right]\alpha^{h,v}\, d(D-Y)$$

where $\mathrm{P}(D-Y)$ denotes the probability density function of the difference between actual and target response $(D-Y)$, to be approximated in section 3.3.

In this approximation, a linearization of the derivative of the energy function with respect to the β^h has been used; the resulting expression is therefore valid only for a limited range. Allowing only relatively small increments ΔMSE in (8), one operates usually in this limited validity range. In (8), the distribution function for $(D-Y)$ must be known; this distribution function is however determined by the total training set and by the total weight vector. A sufficiently accurate estimation of the error distribution function is calculated in section 3.3.

An illustration

Figure 4 illustrates the correspondence between (8) and simulation results for a specific training set. For the calculations, the approximation of the error distribution function as will be described in section 3.3 was used. This approximation of the distribution function requires the ideal eventual *MSE* and the ideal eventual fraction of correctly classified examples. The required $\alpha^{h,v}$ is given in (7); other required parameters for (8) are the type of non-linearity $f(\cdot)$ and η.

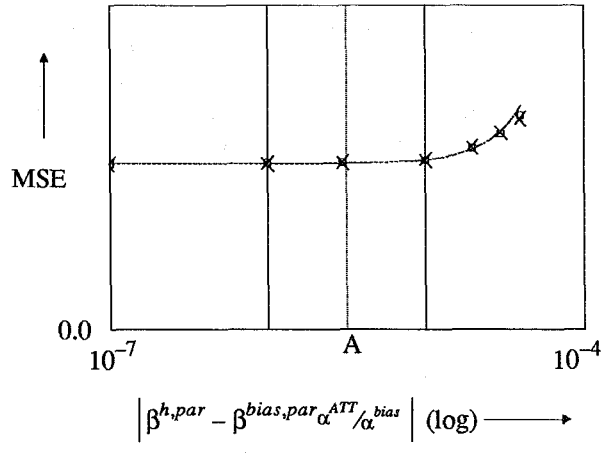

Fig. 4 *Minimum attainable MSE as a function of the $\beta^{h,par} - \beta^{bias,par}\alpha^{ATT}/\alpha^{bias}$; calculation results (lines), and simulation results (□ correspond to negative values, and × correspond to positive values of $\beta^{h,par} - \beta^{bias,par}\alpha^{ATT}/\alpha^{bias}$) ; "A" explained in section 3.2*

3.2 Rule of thumb

It can be derived that the average adaptation of β^h is positive only for linearly separable examples that result in a non-zero weight adaptation, and similarly that all non-linearly separable examples result in an average decrease of β^h. This can be explained as follows (for a more exact description see [1]): the adaptation of β^h on the constant α^{ATT} cancels in first order; adaptation of β^h on the remaining $\alpha^{h,v}$ has a positive sign for linearly separable examples and has a negative sign for non-linearly separable examples.

In the global minimum in the *MSE* versus β^h curve (see figure 2), the total adaptation of β^h is ideally zero. The total adaptation of β^h on linearly separable examples in the global minimum is therefore equal to the adaptation on non-linearly separable examples but positive:

$$\begin{cases} \underline{U}_{Lin.Sep.} & \Rightarrow \overline{\Delta\beta^h} \geq 0 \\ \underline{U}_{Non\_Lin.Sep.} & \Rightarrow \overline{\Delta\beta^h} < 0 \end{cases}$$

The subscript *Lin.Sep.* in these relation corresponds to examples that are linearly separable; input vectors with the subscript *Non_Lin.Sep.* correspond to non-linearly separable training examples. Furthermore, the *average of the adaptation of* β^h must be used in these relations to compensate for constant part in α^h, i.e. to compensate for α^{ATT}.

The relation can be interpreted as follows: in the global minimum, the linearly separable examples generate a "force" which tends to increase β^h and the non-linearly separable examples generate an equal (but with opposite sign) "force" that tends to reduce β^h.

As a rule of thumb, the effect of the parasitic constant weight adaptation is insignificant in case that the ideal adaptation of β^h on the linearly separable examples is at least one order of magnitude larger than the parasitic weight adaptation terms on the right hand side of (6). In formula:

$$\left| \beta^{bias,par} \frac{\alpha^{ATT}}{\alpha^{bias}} - \beta^{h,par} \right| \ll$$

$$\eta \int_{Lin.Sep.} INT(D-Y)\, d(D-Y) \;\Leftrightarrow\; small\ effect \qquad (9)$$

with

$$INT(D-Y) = P(D-Y) \frac{\left(D - f\left(\alpha^{h,v}(D-Y)\,\beta^h\right)\right) f'\left(\alpha^{h,v}(D-Y)\,\beta^h\right)\, \alpha^{h,v}(D-Y)}{P(correct)}$$

$P(D-Y)$ approximated in section 3.3.

The point marked with *A* in figure 3 corresponds to the situation in which the right hand side of (9) is one order of magnitude larger than the left hand side of (9). For the training set used in the simulations, at the point marked with *A*, the increase in MSE is 1% of the optimum *MSE*, while $\beta^{h,end}$ is approximately 0.9 or 1.1 times $\beta^{h,end}_{ideal}$ (depending on the sign of $\beta^{h,par} - \beta^{bias,par}\alpha^{ATT}/\alpha^{bias}$). With (7), equation (9) can be rewritten into

$$\left| \beta^{bias,par} \frac{\alpha^{ATT}}{\alpha^{bias}} - \beta^{h,par} \right| \leq \frac{\eta}{10\, \beta^{h,end}_{ideal}} .$$

$$\int_{Lin.Sep.} P(D-Y) \frac{(D-Y) f'\left(f^{-1}(Y)\right) f^{-1}(Y)}{P(correct)}\, d(D-Y). \qquad (10)$$

Note that the integral in (10) requires only information about the distribution of the error $(D-Y)$, the shape of the non-linear function $f(\cdot)$ and the desired response D. It follows directly from (10) that the maximum tolerable constant weight adaptation decreases linearly with the reciprocal value of $\beta^{h,end}_{ideal}$; i.e. decreases with decreasing distance between the boundaries at which the neuron classifies examples as $Y=D$.

3.3 Approximating the error-distribution

In the mathematical estimation of the maximum of the constant parasitic weight adaptation, the distribution function of the errors $(D-Y)$ is required. An exact description of this distribution function requires knowledge about the total training set and the exact state of the neural network. For estimation purposes, an approximation of the actual distribution function appears to be satisfactory. In this section, it is assumed that the actual distribution function of the errors, $P(D-Y)$ is linearly descending with the error:

$$P(D-Y) = \begin{cases} B - A(D-Y) & (D-Y) < B/A \\ 0 & elsewhere \end{cases} .$$

In this case, it is straight forward to calculate that the constants *A* and *B* are given by

$$A = \frac{(1-P(correct))^2}{3 \cdot MSE} \quad \text{and}$$

$$B = \sqrt{\frac{2(1-P(correct))^3}{3\, MSE}}$$

Although more accurate approximations can be made this does generally not increase the accuracy of the results of the estimations in this paper significantly, while the computational overhead is relatively large.

3.4 Estimation of precision requirements

Assuming that the constant weight adaptation ΔW is identical for all weights, (10) can be used to give a worst case estimation of the maximum tolerable constant weight adaptation:

$$\left| \Delta \underline{W}_i^{par} \frac{\alpha^{ATT}}{\alpha^{bias}} - \Delta \underline{W}_i^{par} \sqrt{N_{in}-1} \right| \approx \frac{\eta}{10\, \beta_{ideal}^{h,end}} \cdot$$

$$\int_{Lin.Sep.} P(D-Y) \frac{(D-Y) f'\left(f^{-1}(Y)\right) f^{-1}(Y)}{P(correct)}\, d(D-Y). \quad (11)$$

The α^{bias} is usually equal to the magnitude of any other element of the input vector $\underline{U}$, for which situation $\left|\alpha^{ATT}/\alpha^{bias}\right| < \sqrt{N_{in}-1}$; where N_{in} is the number of inputs of the neuron. It now follows directly that worst case

$$\Delta \underline{W}_i^{par} \le \frac{\eta}{20 \sqrt{N_{in}-1}\ \beta_{ideal}^{h,end}} \cdot$$

$$\int_{Lin.Sep.} P(D-Y) \frac{(D-Y) f'\left(f^{-1}(Y)\right) f^{-1}(Y)}{P(correct)}\, d(D-Y). \quad (12)$$

Calculations show that the integral in (12) is weakly dependent on the distribution of $(D-Y)$. The integral changes slightly only for extreme performance parameters (either *high* fraction of correctly classified examples and simultaneous a *high MSE* or a low $P(correct)$ and simultaneously a low *MSE*). For a sigmoid non-linearity with unity maximum derivative and a double threshold at 0.95 respectively 0.05 of the maximum neuron response, calculated values for the integral are typically in the range [0.01,0.02]. With these values, a worst case estimation for the maximum tolerable constant weight adaptation is:

$$\Delta \underline{W}_i^{par} \le \frac{\eta}{20 \sqrt{N_{in}-1}\ \beta_{ideal}^{h,end}}\ 0.01 \qquad \text{(worst case)}$$

$$\left(f(x) = \frac{1}{1+exp(-4x)},\ D \in \{0.05, 0.95\} \right). \quad (13)$$

Because the maximum usable adaptation factor η decreases linearly with N_{in} [1], it follows that the worst case tolerable constant weight adaptation scales inversely proportional with $\left(N_{in}\sqrt{N_{in}-1}\right)$. For higher constant weight adaptation values than those indicated by (13), the neuron may learn the training set but the MSE then depends on both the training set (especially on α^{bias} and α^{ATT}) and on the exact values of $\beta^{h,par}$ and $\beta^{bias,par}$.

The value of $\beta_{ideal}^{h,end}$ can easily be derived assuming that the incorrect knowledge part of $\underline{W}$, $\underline{W}^{\varepsilon}$, is small. In this case, $\beta_{ideal}^{h,end}$ is by approximation equal to the norm of the weight vector of the neuron after training, excluding the bias-related weight. By subdividing the eventual weight vector in a bias related weight W^{bias} and in the remaining elements, $\beta_{ideal}^{h,end}$ can be approximated by

$$\underline{W}_{ideal}^{end} = \left[\underline{W}_{ideal}^{end*} \mid W^{bias} \right]^T \Rightarrow \beta_{ideal}^{h,end} \approx \left| \left[\underline{W}_{ideal}^{end*} \mid 0 \right]^T \right|$$

4 An application

In this section, an application of the analysis of section 3 is presented: the minimum size of weight storage capacitors in single-layer neural networks is estimated mathematically. For this estimation, only a few basic parameters are required:

- $[-\hat{W}, \hat{W}]$ is the implemented weight range
- $[-\hat{V}^{weight}, \hat{V}^{weight}]$ is the corresponding weight voltage range (stored on a capacitor)
- ΔQ^{par} is the attainable constant charge injection
- C^{weight} is the weight storage capacitor

It is now straight forward to show that the weight storage capacitors must satisfy

$$C_i^{weight} \ge \frac{\hat{W}\, \Delta Q^{par}}{\Delta \underline{W}_i^{par}\ \hat{V}^{weight}},$$

where the desired accuracy for the weight adaptation, $\Delta \underline{W}_i^{par}$, is given by (12). The largest β^h possible with the specified weight range is $\hat{W}\sqrt{N_{in}-1}$. For a sigmoid non-linearity with unity maximum derivative and a double threshold at 0.95 respectively 0.05 of the maximum neuron response, it follows from (13) that a worst case estimation for the size of the weight storage capacitor is

$$C_i^{weight} \ge 2000 \frac{\hat{W}^2\ \Delta Q^{par}\ (N_{in}-1)}{\eta\ \hat{V}^{weight}} \qquad \text{(worst case)}$$

For $\Delta Q^{par} = 100\ electrons$; $\hat{V}^{weight} = 1.5$; $\hat{W} = 10$; $\eta = 0.1$ and a five dimensional input space (including the bias), the minimum size of the weight storage capacitor is then worst case $80pF$.

Note that depending on the training set, the hardware neural network may learn the training set properly using smaller capacitors because among others the actual $\beta_{ideal}^{h,end}$ depends on the training set. If for example the eventual hyperplane is perpendicular to one of the axes that span the input space and at the same time only half of the weight range is used, the required capacitors need worst case to be $10pF$. Note that in this case the gain of the weight adaptation block is decreased by a factor 8 with respect to the

situation with the $80pF$ weight storage capacitor in order to obtain the same adaptation factor η.
Furthermore, the estimations in this paper are worst-case estimations. As indicated by figure 4, constant weight adaptations up to a factor 3 larger than those indicated by equations (12) and (13) may have a negligibly small effect on the attainable performance. Therefore, the weight storage capacitors may be taken about a factor 3 smaller than indicated by the worst-case estimations. For even smaller weight storage capacitors, the neural network may learn properly, but the difference between the eventual performance and the optimum performance will depend heavily on the training set.

5 Conclusions

This paper presents a mathematical estimation of precision requirements for analog weight adaptation circuitry for single-layer feed-forward neural nets. It is shown that for a specific non-linearity and specific threshold values, the worst-case precision depends only on the adaptation factor, the eventual norm of the weight vector and on the dimension of the input space. For precisions lower than those indicated by the worst-case estimations in this paper, the neural network may learn the training set, but the eventual performance will be heavily dependent on both training set and on the precisions of all weight adaptation blocks. With the precision estimations in this paper, one can estimate whether given analog hardwired on-chip learning neural networks are feasible for given training sets. Another application of the analysis in this paper is the estimation of the minimum size for weight storage capacitors.

6 References

[1] A.J.Annema, *Analysis, Modelling and Implementation of Analog Integrated Neural Networks*, Ph.D. Thesis, University of Twente, The Netherlands, 1994

[2] A.J.Annema, K.Hoen, and H.Wallinga, "Learning Behavior and Temporary Minima of Two-Layer Neural Networks", accepted for publication in *Neural Networks*, 1994

[3] D.D.Caviglia, M.Valle, and G.M.Bisio, "Effects of Weight Discretization on the Back Propagation Learning Method: Algorithm Design and Hardware realization", *in Proceedings of the IJCNN 1990*, San Diego, vol.2, pp. 631-637

[4] D.D.Caviglia, M.Valle, and G.M.Bisio, "A CMOS Analog Architecture for Adaptive Neural Networks", in *Silicon Architectures for Neural Nets*, eds. M.Sami and J.Cadzadilla-Daguerre, 1991, pp. 113-123

[5] M.Valle, D.D.Caviglia and G.M.Bisio, "An Experimental Analog VLSI Neural Chip with On-Chip Back-Propagation Learning", in *Proceedings of the ESSIRC 1992*, Copenhagen, pp. 203-206

[6] Y.Wang, "A Modular Analog CMOS LSI for Feedforward Neural Networks with On-Chip BEP Learning", in *Proceedings of the IEEE ISCAS 1993*, Chicago, pp. 2744-2747

[7] A.J.Montalvo, P.W.Hollis, J.J.Paulos, "On-Chip Learning in the Analog Domain with Limited Precision Circuits", in *Proceedings of the IJCNN 1992*, Baltimore, vol. I, pp. 196-201

[8] T.Lehmann, "A Hardware Efficient Cascadable Chip Set for ANN's with On-Chip Backpropagation", *International Journal of Neural Systems*, vol. 4, 1993, pp. 351-358

[9] R.Gregorian, and G.C.Themes, *Analog MOS Integrated Circuits for Signal Processing*, New York: Wiley, 1986

[10] D.E.Rumelhart, G.E.Hinton and R.J.Williams, "Learning internal representations by error propagation", in *Parallel Distributed Processing*, vol. 1, Cambridge, MA.:MIT Press, 1986