Power Aware Learning for Class AB Analogue VLSI Neural Network

Sankalp S. Modi, Peter R. Wilson, Andrew D. Brown
School of Electronics and Computer Science
University of Southampton
Southampton, UK
{ssm03r,prw}@ecs.soton.ac.uk

***Abstract*—Recent research into artificial neural networks (ANN) has highlighted the potential of using compact analogue ANN hardware cores in embedded mobile devices, where power consumption of ANN hardware is a very significant implementation issue. This paper proposes a learning mechanism suitable for low-power class AB type analogue ANN that not only tunes the network to obtain minimum error, but also adaptively learns to reduce power consumption. Our experiments show substantial reductions in the power budget (30% to 50%) for a variety of example networks as a result of our *power-aware learning*.**

I. INTRODUCTION

The compact size and low power dissipation of analogue ANN has made it an attractive choice for hardware and it has attracted considerable research efforts in recent years [1;2]. As specialized ANN hardware finds many potential applications in mobile embedded devices, the power consumption becomes a major issue [3]. Since shrinking biasing voltages makes it difficult to process high resolution data in voltage-mode, there has been increasing emphasis on the low power current mode (CM) implementation of the ANN [4], which gives better results at lower bias. The Class AB CM implementations are particularly attractive options as they remove the necessity to maintain large bias current levels (leading to very low-power consumption) and this allows the input signal magnitude to exceed the bias current (improving calculation precision)[3].

The power consumption of such class AB CM ANN depends heavily on the values of signal currents, which in turn depend on the values and distribution of weights of synaptic connections. Since the weights are determined by the applied learning process, the learning process is very likely to affect the power consumption considerably. To the best of our knowledge, none of the currently used ANN learning algorithms are capable of taking into account of the effect of the weight distribution on the power consumption. We propose a new power-aware learning algorithm that is sensitive to the power consumption of the design. The algorithm is based on the variation of weight perturbation [5] algorithms; it adds a penalty term for power consumption in the objective function. We have applied our algorithm on a sample class AB ANN described in [4] for various classification and function approximation tasks. The results of these experiments are discussed in this paper.

II. POWER-AWARE LEARNING

A. Complexity Regularization with Penalty-term

An essential aspect of neural network training is to improve generalization. A class of commonly used techniques for this are known as complexity regularization, which aims to prevent the learning algorithm from over-fitting the training data by restricting the complexity of the ANN function. A popular approach is to include an additional penalty-term in the cost function of learning algorithms, which penalizes overly high model complexity(also known the *penalty term pruning*) [6;7].

$$O(\mathbf{w}) = E(\mathbf{w}) + \lambda_c \cdot C(\mathbf{w}) \qquad (1)$$

$O(\mathbf{w})$ is the objective function that is to be minimized with respect to weight vector $\mathbf{w}$, the vector of synaptic weights. $E(\mathbf{w})$ is the error function, usually the Mean Squared Error (MSE) over the training samples. $C(\mathbf{w})$ is the complexity penalty term. λ_c, the regularization parameter determines the influence of the complexity penalty on the learning procedure.

B. On-chip Learning and Weight Perturbation

On-chip learning can greatly increase the training speed and realize the full potential of the massive parallelism of analogue VLSI ANN. Moreover, on-chip implementation of a learning mechanism is required for adaptive systems.

Traditional error back-propagation approaches require high precision calculations and precise modelling of the activation function, which are unsuitable for on-chip implementation in analogue VLSI. Alterative Weight

School of Electronics and Computer Science, University of Southampton, UK*(sponsors)*

Perturbation methods have been developed [2;5] and implemented successfully on analogue/mixed mode VLSI. In these methods, the effect of random weight perturbations on output error is observed and the gradient information is *measured* rather than *calculated*, thus avoiding the complicated derivative calculations and backward error propagation. These techniques do not assume any model for implemented ANN and hence networks can learn to compensate for analogue circuit non-idealities [2].

C. Power-aware Weight Perturbation

As explained in section I, the power consumption of class AB CM ANN system heavily depends on the values and distribution of weights. For power-aware learning in such systems, we propose an alternative objective function as presented in (2).

$$O(\mathbf{w}) = E(\mathbf{w}) + \lambda_p \cdot P(\mathbf{w}) \qquad (2)$$

$P(\mathbf{w})$ is the penalty term for power consumption during the feedforward phase and λ_p is the power regularization parameter. (In most of the practical ANN hardware applications, the utilization period of the trained ANN is much larger compared to the training period. Hence, we assume that only the power consumption of the feedforward phase is significant.) There are several difficulties in implementing this power aware learning with standard Back-propagation offline training. First, the effect of weight vector $\mathbf{w}$ on power (i.e. $P(\mathbf{w})$) is difficult to estimate. In addition, back-propagation is based on the calculation of gradient of the objective function with respect to weight w_i. The expression for partial derivative of the power term ($\partial P(\mathbf{w})/ \partial w_i$) cannot be defined precisely, making it unsuitable for standard back propagation calculations.

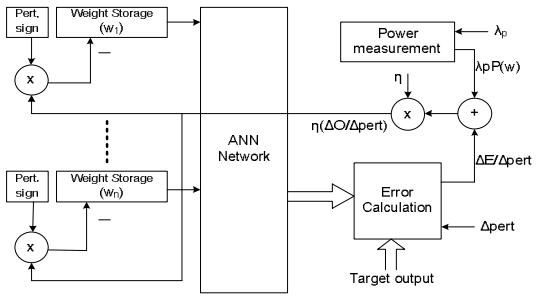

Figure 1. Power-aware weight perturbation learning implementation

However, in an on-chip learning scenario, the power consumption can be easily measured and used with the measured error to form a new objective function for a power-aware weight perturbation learning scheme. Since the weight perturbation learning is 'model-free' and it is driven solely by the measured objective function, this modification in the objective function does not need any extra calculations and can be implemented with minimal overheads as shown in Figure 1. With the addition of the power-penalty term, we are essentially attempting a multi-objective learning problem. Although, there have been previous attempts to implement multi-objective learning and regularization in ANN, to the best of our knowledge, none of the attempts involves reduction in power consumption as one of the objectives.

III. EXPERIMENTS AND SIMULATION RESUTSL

A. Class AB ANN Implementation and Estimate of Power Consumption

For the experiments, we have considered the low power class AB CM ANN analogue cells presented in [3;4]. Current consumption and power consumption of the multiplier cell and the sigmoid activation function cell is approximated in equations 3 and 4 respectively [4]. (For the multiplier cell, current i_s represents synaptic weight and current i_{in} is incoming current from the previous layer) The total power consumption of ANN is assumed to be the sum of total power consumption of all the multiplier and activation units.

$$I_{mult} \approx 0.5 \cdot i_s + 2.5 \cdot i_{in}, P_{mult} \propto I_{mult}^2 \qquad (3)$$

$$I_{act} \approx 2 \cdot i_{in}, P_{act} \propto I_{act}^2 \qquad (4)$$

This approximation of power consumption is not very accurate and accurate power consumption can be a more complex non-linear function of input currents, especially at low signal levels. However, our proposed on-chip learning is driven by the *actual on-chip power measurement* and does not require evaluation of any equation to obtain power consumption. Hence the choice of the power approximation function is not critical for our experiments and for the purpose of demonstration, we have used (3) and (4) for the simulation experiments described in this paper. Indeed, our additional experiments show that adding small non-linearity and offsets to (3) and (4) does not alter our results significantly.

B. Experiments and simulation results

1) The 8-3-8 encoder problem[8]

The graph in Figure 2 shows the training results for 8-3-8 encoder problem, which is a widely used example for ANN benchmarking.

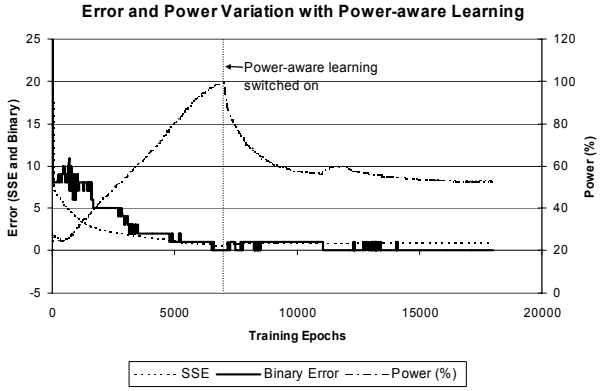

Figure 2. Power-aware learning for 8-3-8 Encoder problem

ANN contained (8+3+8=) 19 neurons with bipolar sigmoid activation function. Adjacent layers were fully connected and no shortcut connections were allowed. Weight vector contained =59 elements (48 connections weights plus 11 biases). Error was measured as the Sum of Squared Error (SSE). Percentage power was measured with respect to the maximum power during the entire training period. All the weights were initialized with zero and the weights were restricted within the interval of [-10,10]. The training was performed with the parallel weight perturbation with power penalty term as per (2) in batch update mode. All simulation experiments in this paper were simulated on Stuttgart Neural Network Simulator (SNNS). As we can see from the graph in Figure 2 that after switching on the power-aware learning, power is considerably (47%) reduced without increasing the error.

2) Proben1 benchmark dataset

We also carried out experiments on the number of problem available in the Proben1 benchmark datasets [9] (various real-life classification and approximation tasks). The experiments where performed on the 'pivot architecture'[9] for each problem with shortcut connections. Precision of all the calculations and weights were restricted to 0.001 to reflect the limited precision available in the analogue hardware, and the weights and calculation results were scaled and restricted within the interval of [-10,10] to reflect the limitation imposed by limited operating range of the transistor devices.

Each network was first trained to achieve minimum validation error with Parallel Weight Perturbation without the power penalty term and minimum Mean Squared validation Error(MSE_{min_valid}) achieved was recorded. Those trained networks were then further trained with power-aware learning such that its Mean Squared validation Error(MSE_{valid}) did not exceed above 5% of MSE_{min_valid} (i.e. $MSE_{valid} < 1.05* MSE_{min\_valid}$). The results of the achieved power reduction for each problem are presented in Table 1. (further details of each benchmark problems can be found in [9]). It can be seen from the table that our proposed approach achieves significant power reduction in a variety of complex problems without increasing error.

TABLE I. RESULTS OF POWER –AWARE LEARNING ON THE PROBEN1 BENCHMARK PROBLEMS

Problem Type	Dataset	Architecture	Power reduction (%)
classification	Cancer	9+8+4+2 L	29.3
Func. Approx	Builing	14+16+8+3 L	26.2
Func. Approx	Flare	24+32+3 L	28.3
classification	Glass	9+16+8+6 S	48.8
classification	Diabetes	8+16+8+2 S	39.5
classification	Thyroid1	21+16+8+3 S	30.7

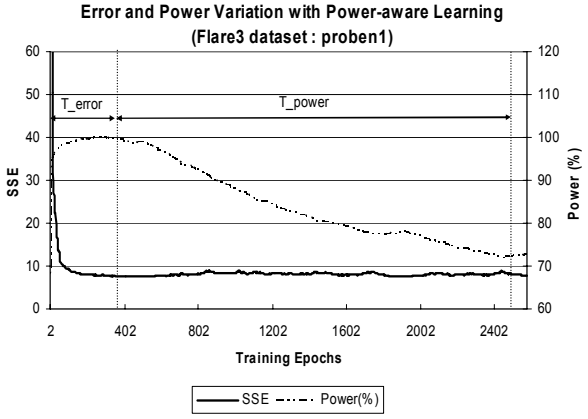

Figure 3. Training times for Flare3 dataset in proben1 benchmark

C. Observations

1) The power regularization parameter λ_p

The algorithm is quite sensitive to the value of λ_p and it is difficult to tune. We have tried number of different strategies to set λ_p. The most successful strategy amongst our experiments was to first train network with $\lambda_p = 0$ to obtain minimum validation error and then slowly increasing λ_p. This strategy is similar to the complexity regularization strategies described in [7]. When the training was stared with non-zero λ_p , ANN was generally unable to reduce the error to the level with the ANN trained with $\lambda_p = 0$, even if later in the training λ_p is reduced to zero

2) Training time

The training time required to achieve low-power (*T_power)* is typically much larger in comparison with the training time to reduce the error (*T_error*). (i.e. The Low-power objective is achieved at a much slower rate in comparison with the Low-error objective.) Figure 3 shows result of the training in Flare3 dataset in Proben1 Benchmark. This is not surprising, because initially with $\lambda_p=0$, learning has only a single objective; while after the power-aware learning is switched-on, the network is attempting more complex multi-objective learning. For the problems attempted from Prben1 benchmarks, the ratio *T_power*/ *T_error* is typically 5-10 or greater. This indicates that that for the ANN applications requiring relatively quick adaptations, our proposed approach may not be able to yield significant power saving due to insufficient training time.

3) Issue of generalization capability

ANN can lose the capability of generalization due to overtraining. Generally 'early stopping' is used to prevent overtraining of ANN [6]. Since power-aware learning requires a considerably large number of training epochs even after early stopping point, there is a danger that we might over-train the network for the training dataset and lose its generalization capability. However, since Power-aware learning generally restricts the free network parameters (i.e. weights) to small values, it acts as a form of complexity regularization mechanism and hence prevents the loss of generalization. In all our experiments, we found that

additional training with power-penalty term did not degrade the generalization performance in any of the problems and in many cases, it actually improved the generalization. Please note that the results presented in Table 1 are obtained within the tight constrains of the validation dataset error and not the training dataset error, which indicates that the networks maintained their generalization capability with Power-aware training.

4) Update style

Parallel weight perturbation with 'update-by-pattern' learning approach is generally considered better in terms of learning speed in comparison with 'update-by-epochs'[2]. However, when 'update-by-pattern' was applied with power-penalty term for power-aware learning, it produced inferior results.

D. Comparison with the weight decay regularization scheme

In class AB CM ANN design, the lower values of weights is likely to consume low power due to the direct relationship between signal levels and power consumption. Hence back propagation learning with a weight decay[6] complexity regularization mentioned in section 2.1 can also drive such circuits towards lower power consumption. With this in mind we need to consider the potential advantages of using the suggested power-aware weight perturbation in comparison with the weight decay regularization. Power-aware weight perturbation has several appealing aspects:

1) Weight decay using (1) is basically aimed at the complexity reduction for improved generalisation and not for power reduction. The relation between P(w) and C(w) can be highly nonlinear, especially with the non-linearities and offsets involved at the very low weights in analogue VLSI.

2) On-chip implementation of the weight decay mechanism is costly in terms of hardware as it requires an additional multiplication for *each weight*. Moreover, the limited precision available in the analogue VLSI can be a major limiting factor for implementing an on-chip weight decay scheme. The implementation of our proposed power-aware learning requires a single power measurement unit for the whole ANN (Figure 1) and does not require very high precision calculations.

3) There is a fundamental difference between the approaches. Weight decay procedure treats all weights in Multi-Layer Perceptron (MLP) equally which is not an appropriate strategy for power reduction because the power consumption not only depends on the weights but also on the input patterns. Learning algorithm should adjust weights in order to prevent high value signal propagation. Unlike our learning process, the weight *decay procedure is incapable of applying preferential treatment according to the input patterns* to different weights in order to achieve better power reduction. We tried to reduce the power consumption using weight decay in a few Preben1 problems and the results presented in Table 2 show that weight decay regularization provides inferior results in comparison to the proposed Power–aware learning.

TABLE II. WEIGHTS-PERTURBATION LEARNING WITH WEIGHT DECAY

Problem Type	Dataset	Architecture	Achieved Power reduction (%)
classification	Cancer3	9+8+4+2 L	< 2
Func. Approx	Builing3	14+16+8+3 L	< 3
Func. Approx	Flare1	24+32+3 L	4.22

IV. CONCLUSIONS

In this paper, we have proposed a novel power-aware learning mechanism for class AB analogue neural network VLSI which is suitable for on-chip implementation. Experiments on the standard Proben1 benchmark problems indicate that it is capable of significant power reduction over a wide range of problems. Key observations on training time, regularization parameter and issue of generalization were discussed. The proposed algorithm shows significant advantages over the other possible low power training method i.e. weight decay regularization. The implications of this work are that an on-chip implementation could lead to significant benefits for practical ANN applications in mobile embedded devices.

REFERENCES

[1] S. Draghici, "Neural networks in analog hardware-design and implementation issues," *International Journal of Neural Systems*, vol. 10, no. 1, pp. 19-42, Feb.2000.

[2] M. Valle, "Analog VLSI implementation of artificial neural networks with supervised on-chip learning," *Analog Integrated Circuits and Signal Processing*, vol. 33, no. 3, pp. 263-287, Dec.2002.

[3] K. Wawryn and A. Mazurek, "Low power, current mode circuits for programmable neural network," in *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196),* vol. 2 ed Sydney, NSW, Australia: IEEE, 2001, pp. 628-631.

[4] K. Wawryn and A. Mazurek, "Low power programmable current mode circuits," *Analog Integrated Circuits and Signal Processing*, vol. 36, no. 1-2, pp. 119-136, 2003.

[5] M. Jabri and B. Flower, "Weight perturbation: an optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks," *IEEE Transactions on Neural Networks*, vol. 3, no. 1, pp. 154-157, Jan.1992.

[6] S. Hyakin, *Neural Networks: A Comprehensive Foundation*, 2nd Edition, ed Prentice Hall, Upper Saddle River, New Jersey 07458, 1999.

[7] C. Jutten and O. Fambon, "Pruning methods: a review," in *3rd European Symposium on Artificial Neural Networks ESANN '95. Proceedings* Brussels, Belgium: D facto, 1995, pp. 129-140.

[8] S. E. Fahlman, "An Empirical Study of Learning Speed in Back-Propagation Networks,"CMU-CS-88-162, 1988.

[9] "Proben1- A Set of Neural Network Benchmark Problems and Benchmark Rules," 1994, ftp://ftp.ira.uka.de/pub/papers/ Techreports/1994/1994-21.ps.