An Analog Artificial Neural Network Design Kit for General Applications

Ansgar Wego, Gundolf Geske, Frank Stüpmann

1 Introduction

The benefits of hardware integrated artificial neural networks (ANN) in terms of multilayer perceptrons (MLP) are well known. Particularly VLSI ANNs easily outclass software implemented neural networks regarding response speed and compactness. But the most challenging task designing analog hardware ANNs is to adapt the synaptic weights to the desired network function.
In this paper, an ANN design kit for general applications using a CMOS integrated ANN chip (Silimann®) and a two-step training technique (an "Off-Chip" and "Loop" step) will be introduced. In the first step of this training procedure, the network weights are determined Off-Chip by means of a model based backpropagation adaptation algorithm. In the second step, the chip is put into a loop and a fine tuning of the weights is done via a stochastic error gradient descent adaptation algorithm (weight perturbation [2]).

2 Model based Off-Chip training

The feed forward phase of neural network processing is rather simple. The input signals i are weighted by the synapses w and summed subsequently. The neurons transmit the weighted sum signals with a (gen. sigmoidal) transfer function. This is performed in every layer of the network.

$$o = f\left(\sum w \cdot i\right) \tag{1}$$

An error E is calculated from the difference of the network output o and the desired output d

$$E = d - o\,. \tag{2}$$

In order to find the global error minimum, gradient descent training methods require the calculation of the error gradient. In terms of backpropagation [1], the error function must be differentiated with respect to the synaptic weights.

$$min = \frac{\partial E}{\partial w} \tag{3}$$

As a consequence, it is necessary to differentiate the output signals of every layer (except the input layer).

2.1 Network model

2.1.1 Neuron circuit

The task of a neuron is to perform a non linear mapping of an input signal to an output signal. Therefore, sigmoid shaped squashing functions are applied. In software implementations, the hyperbolic tangent or the logistic function are commonly used.

To obtain a sigmoid like transfer behavior of a hardware implemented neuron, a source-coupled MOSFET pair is used. For this circuit a sufficient approximation is [3]

$$I_d(V_d) = kV_d\sqrt{\frac{2I_{ss}}{k} - V_d^{\,2}} \tag{4}$$

where k is given by the fabrication process and the design parameters (W/L)

$$k = \mu_0 C_{ox} \frac{W}{2L}$$

I_d is the differential output current. I_{ss} is the source current and V_d is the differential input voltage.

2.1.2 Synapse circuit

The synapse modulates its input signal with a stored weight by simply multiplying both signals. This function is performed by a Gilbert multiplier using two source-coupled MOSFET pairs in parallel and taking the difference of the output currents. Thus, the approximation yields

$$I_d(I_{in}, V_d) = k \cdot V_d \left(\sqrt{\frac{2(I_{ss} + I_{in})}{k} - V_d^2} - \sqrt{\frac{2(I_{ss} - I_{in})}{k} - V_d^2} \right) \tag{5}$$

This equation assumes identical parameters k for all concerned MOSFETs. I_{in} modulates the source current and it is the first input signal, while the differential voltage V_d (from a weight storage capacitor) is the second signal for the multiplication task of the synapse.

2.1.3 Backpropagation part requirements

The above introduced equations were applied to the feed forward part of an off-chip training software to obtain a relatively real model of the hardware. For the backpropagation part, some approximated functions are necessary to get reasonable training results. This is due to the properties of the neuron's transfer function Equ. (4). The function is physically sensible only for

$$-\sqrt{\frac{I_{ss}}{k}} < V_d < \sqrt{\frac{I_{ss}}{k}}\,.$$

Outside this range, I_d takes on constant values (i.e. I_{ss}). The derivative yields zero values outside the given range and sharp transitions at its bounds. To apply backpropagation, it is more practical to use an approximation for the derivative of Equ. (4). An appropriate function with soft transitions and non-zero values is given in Equ. (6)

$$\frac{\partial I_d(V_d)}{\partial V_d} \approx \sqrt{2 I_{ss} k} \cdot \left(1 - \left(\tanh\left(\sqrt{\frac{2k}{I_{ss}}} \cdot V_d \right) \right)^2 \right). \tag{6}$$

3 Loop training

The weight values that have been determined off-chip typically lead to a proper network operation after having been loaded into the chip. Particularly if simple classification tasks have been trained, the differences between the network model and the real chip network are negligible. But if precise approximation tasks have to be performed, the only way to get a specific weight set is to apply a post training with the real network chip. In this post training the chip is put into a loop. Analog input and output PCI cards are used to link the chip and a computer. The patterns p of the training set are presented in sequence to the chip inputs. The outputs are sampled and the error between the desired output d and the measured output o is calculated (cf. Fig. 1).

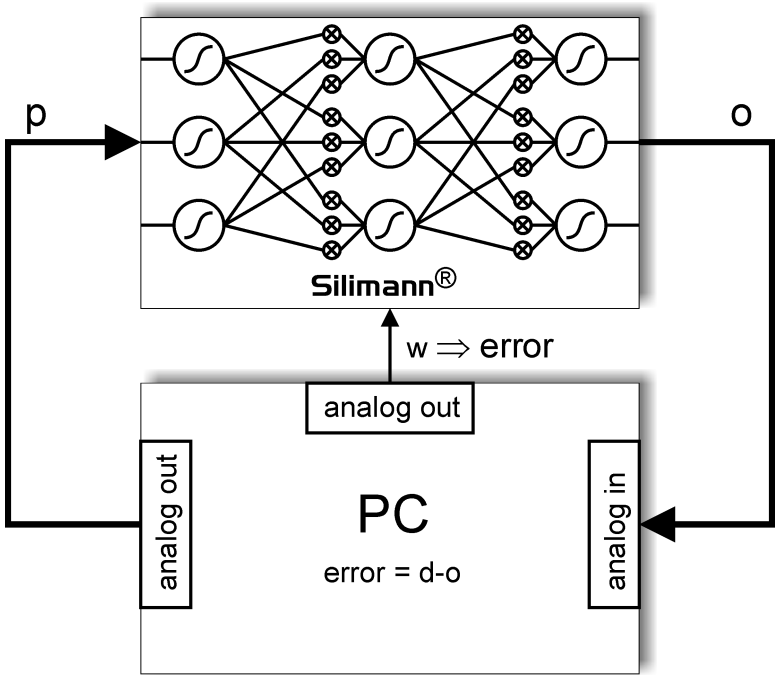

Fig. 1: Loop training principle of a multiplayer perceptron network

The error signal *E* is used to determine the weight changes. The loop training method used is a stochastic error gradient descent algorithm also known as weight perturbation. At this algorithm, only approximations of the error gradient are applied (Equ 7).

$$min = \frac{\Delta E}{\Delta w} \tag{7}$$

4 Training results

The VLSI ANN chip Silimann®120cx, which has to be trained by the developed software, is a fully connected two layer small-sized analog feed forward network with 10 input, 6 hidden, 10 output and 2 bias neurons. A total number of 136 synapses are implemented. This chip is designed for sensor signal processing purposes.

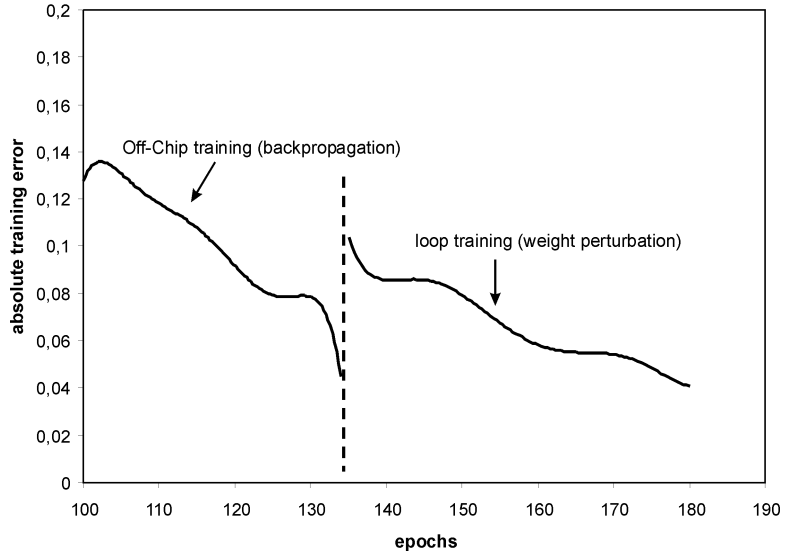

Fig. 2: Typical two-step training curve

A common application for the Silimann®120cx is color signal classification. In this application, a RGB sensor delivers three signals (i.e. red, green, blue) for a distinctive color. These signals form the input vector. For every input vector (color signal) an unique output vector must bee assigned to form input-output pairs. These patterns in the first step are trained off-chip on a computer. The resulting synaptic weights are downloaded into the chip subsequently.

In the second step the loop training is performed. Fig. 2 shows a typical curve of the training behavior. As can bee seen in the figure, after downloading the weights at first the error is bigger. But due to the loop training, the real network weights are adapted properly and the error gets smaller.

5 Conclusions

The results of performance tests show that despite statistical circuit tolerances, weight sets which were adapted using the off-chip training method and downloaded subsequently into the chip, are typically sufficient to get functional networks. This is possible due to a model based training process. In cases where precision is needed, a subsequent loop training improves the results obviously.

The main benefit of applying the discussed two-step training procedure on the one hand is getting the backpropagation algorithm utilized for hardware implemented ANNs. In this way, the advantages of backpropagation (convergence speed, applicability etc.) are available.

On the other hand it is feasible to bypass the hardware problems of analog ON-Chip training techniques [4]. With a short post training using the discussed loop method it is possible to eliminate circuit imprecisions which arise due to fabrication tolerances.

6 References

[1] D. Rumelhart, J. McClelland: "Parallel Distributed Processing: Explorations in the Microstructure of cognition" Vol. 1, MIT Press, Cambridge, MA, 1986

[2] M. Jabri, B. Flower: "Weight Perturbation: An optimal Architecture and Learning Technique for Analog VLSI Feedforward and Recurrent Multy-Layer Networks", IEEE Trans. Neural Networks, Vol. 3 (1), pp. 154-157, 1992

[3] P. R. Gray, R. G. Meyer: " Analysis and Design of Analog Integrated Circuits", J. Wiley & Sons 1993

[4] B. K. Dolenko, H. C. Card: "Tolerance to Analog Hardware of ON-Chip Learning in Backpropagation Networks" IEEE Transactions on Neural Networks, Vol. 6, NO.5 Sep. 1995

7 Authors

Dr.-Ing. Ansgar Wego
Dr.-Ing. Gundolf Geske
Dr.-Ing. Frank Stüpmann

Silicann Technologies GmbH
Joachim-Jungius-Str. 9, 18059 Rostock
www.silicann.com