Chapter 3

Generalized Artificial Neural Networks (GANNs)

In this chapter, following an introduction of different models used for ANNs, we describe a general model for neural networks with localized storage of parameters. Correspondingly, as a special case covered by this model, we introduce a quadratic synaptic relation, similar to that found in practical MOS devices, as a means by which to implement synapses in ANNs. A simulator has been developed to account for the activity of this kind of synaptic unit and to account for the additional constraints implied by bounded weights. This simulator and the training equations for one particular version of simple MOS-compatible synapses are discussed next, and then several conventional test problems used in this work are introduced. Finally, the results of our simulations are shown, and the necessity for a more detailed analysis of networks composed of simple-transistor synapses is discussed.

3.1 Introduction

Neural-network researchers have introduced many novel techniques and algorithms at the software and system levels. In this development, many new directions for information processing have been promoted [4]. In the whole process, the concepts of training and learning have established their place beside traditional techniques of programming. Machines with the capability of generalization and the potential to reach a reasonable solution when faced with never-seen cases are now familiar. Associative memory and unsupervised data clustering are possible and accepted concepts. As a result, the software and algorithmic research communities have been able to advance and thereby develop many new concepts. Now, the time has come when one might ask the question: “Should the hardware-research
community expect the appearance of similar improvements and the introduction of new techniques and concepts in the neural-network-implementation field? “

In the past few years, much effort has been directed towards VLSI implementation of ANNs [15]. Many problems have been revealed and some of them are partially solved. Currently low power consumption [56] and availability of accurate analog computational techniques [73] are issues under investigation. Noise and offset reduction through the process of adaptation now has been shown to be possible [51]. The capability of the training process to extract knowledge embedded in the input data has already been demonstrated. Further, the hardware community has shown that it is also possible to exploit a training process to characterize the processing hardware itself [51]. Moreover, it has been demonstrated that the functionality of a basic MOS device can be increased to correspond to that of a simple neuron with constant synaptic weights [67].

In our view, in the pursuit of improved implementations, we should not necessarily try to copy biological models; for, in comparison to silicon-based ANNs, it is clear that the media, the available resources, and the applications are all different. Thus, to progress, we define a generalized analog neural network as a parallel storage-processing mechanism of some kind, capable of memorizing, retrieving, and updating organized knowledge in an analog form (including, for example chemical deposition of certain ions in biology, as well as storage of a voltage, or a charge, or possibly a current, in an analog artificial neural network). This storage-processing medium should also be capable of reacting to its inputs as they originate in the environment or in other neighboring blocks, and from them to provide an output. Based on the above perceptions, we have tried to re-evaluate the available hardware resources in conventional CMOS/BiCMOS technologies and to provide an implementation based on optimal usage of these resources. Our early attempts are reflected in [74], where we discuss the possible deployment of the characteristic equation of a MOS transistor as the basic synaptic operation of an ANN.

In our approach, we try to maintain the most important aspect of neural computational theory, which we view to be that of simple interconnected adaptable processing elements operating in parallel. As to the issue of how to implement basic blocks and their interconnects, instead of following the conventional approach of compiling software blocks into hardware, we have tried to exploit hardware resources as they naturally exist, especially in the form of MOS transistors embedded in a CMOS technology. From the users’ point of view, the outcome may be quite the same as with the conventional approach. However, it is possible that internal processing based upon different internal implementations for interconnects and processing elements can lead to a much more efficient design. In this chapter, our early attempts to explore some new hardware-based synaptic operations for ANNs are detailed.