5. Artificial Neural Networks

5.1 Overview

Learning algorithms are required to operate in ill-defined and time-varying environments with a minimum amount of human intervention. These techniques are typically used to control plants for which a conventional mathematical analysis is not possible, and many different learning systems have been proposed for use within IC systems. Artificial Neural Networks (ANNs) are biologically inspired learning algorithms and control architectures. Several neural controllers were developed in the sixties, most notably Widrow's inverted pendulum controller in 1963 [Widrow, 1987]. The development of new network architectures and learning algorithms has stimulated control engineers to re-evaluate the potential of ANN-based controllers, or *neuro-controllers*, and many of the learning algorithms have their parallels in the adaptive identification and control fields.

ANN developments reflect the desire to develop alternatives to instruction-based von Neumann computing. Humans perform many tasks (speech and image recognition, complex coordination and control tasks) with relative ease that are very difficult to solve using traditional algorithmic computing techniques. The brain's architecture is vastly different from the commonly used serial computer, and neural researchers aim to endow machines with human-like information processing capabilities.

Present neural-like computational architectures are based on a simplified model of the brain, with the processing tasks being *distributed* across many *simple* nodes (or neurons). The power of these algorithms comes from the collective behavior of the simple nodes. Each ANN is completely specified (modeling and learning abilities) once the network topology, the transfer function of each node and the learning rule have been determined.

5.2 Neural Computing Benefits

Neural computing is different from conventional algorithmic computing, although the former can generally be decomposed into an algorithm and implemented on a serial machine. These apparently contradictory statements can be resolved if it is accepted that it is the *approach* which distinguishes the two techniques, rather than the final implementation. Neural networks offer solutions to problems that are very difficult to solve using traditional algorithmic decomposition techniques and the potential benefits of a neural approach are:

- <u>learning</u> from the interaction with the environment, rather than by explicit programming;
- <u>few</u> restrictions are placed on the type of functional relationship that can be learned;
- <u>ability</u> to generalize (interpolate and extrapolate) the training information to similar situations; and
- <u>inherently</u> parallel design and the computational load can be evenly distributed across many simple processing elements. Thus, the network possesses some degree of fault tolerance with respect to processor failures.

The first three properties are desirable for any learning algorithm, and the fourth can be used to apply these networks to larger real-time systems. If a learning algorithm possesses these properties, it can yield the control system with the following advantages [Stengel, 1992]:

- <u>decreasing</u> the required amount of human intervention;
- <u>increasing</u> the flexibility of the control system;
- <u>improving</u> the performance of the control system; and
- <u>reducing</u> the initial design time and cost.

The performance of an ANN (learning, recall, computational burden, etc.) depends on how well it satisfies the first property list, and this determines its potential for off-line design problems. For on-line adaptive modeling and control, the algorithm also needs to possess the following properties:

• learn significant information in a stable manner and in real-time;

and

• provable learning convergence and stability properties.

Some of the most commonly used ANNs satisfy neither of these properties, although this means that while they can be used in neurocontrol applications, learning should only occur off-line.

5.3 Neural Network Terminology

The ANNs consist of a large number of simple processing elements called *nodes*. Signals are passed between nodes along weighted connections, where the *weights* are the network's adjustable parameters. The arrangement of the network's nodes and connections defines its architecture and there are many possible variations. One typical arrangement is the case where the nodes are arranged into layers and each node in one layer has connections only with nodes in the preceding layers. The input is applied to the first layer and this information is propagated *forwards* through the network. The signals at the final layer are then the network's output. *Feedback* or *recurrent* networks have additional connections where the output of at least one node is propagated backwards as input to the nodes in the same layer [Goles and Martinez, 1990, Ku and Lee, 1995].

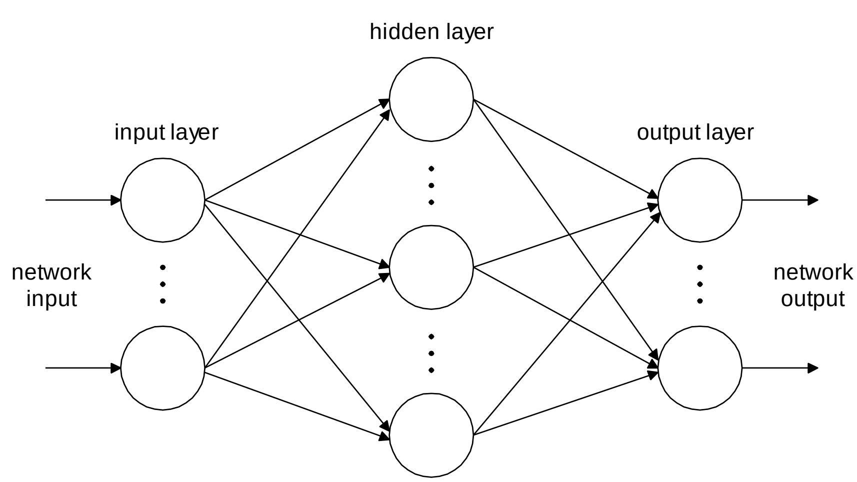

Fig. 5.1. A feedforward multi-layer network where each circle corresponds to a node and each

arrow represents a weighted link.

The learning rules used to train the networks can generally be classified as *supervised* or *unsupervised*. Supervised learning rules require the desired network output to be available and they adapt the weights so that the output approaches to the desired one [Widrow and Lehr, 1990].

Unsupervised learning is used to organize the network's structure based only on the training inputs presented to the network. [Kohonen, 1990]. This type of algorithm can be used to develop representative data set features, for vector quantization and dimensionality reduction clustering.

5.4 Historical Perspective

Widrow and Smith developed one of the first neurocontrollers [Widrow, 1987], which is a simple ADAptive LINear Element (ADALINE) taught to reproduce a switching curve in stabilizing and controlling an inverted pendulum. This ADALINE was one of the first ANNs along with the Rosenblatt's Perceptron [Rosenblatt, 1961] and has a simple architecture used extensively in other ANNs [Miller *et al.*, 1990c]. The output of neurocontroller is discrete and can be used to represent binary control actions.

Albus proposed the Cerebellar Model Articulation Controller (CMAC) during the seventies as a tabular model of the functioning of the cerebellum and used it to control robotic manipulators. The CMAC has been used to model and control nonlinear chemical processes [Tolle and Ersü, 1992] and robotic applications [Miller *et al*., 1990b]. The CMAC was derived from the Perceptron's architecture by combining it with the ADALINE, where a binary encoding of the input space is used. It could be viewed as a modified ADALINE network which generates a pseudo-continuous output.

Many different ANNs and IC architectures were developed during the eighties. Reinforcement learning and adaptive critic schemes have been developed [Miller *et al.*, 1990c] and ANNs such as the Multilayer Perceptrons (MLPs) [Rumelhart and McClelland, 1986], Radial Basis Functions (RBFs) [Chen and Billings, 1994], Functional Link Nets (FLNs) [Pao *et al.*, 1994] and B-splines [Moudy, 1989] have been developed. Recurrent networks have been used in optimization [Hopfield and Tank, 1986, for plant modeling and estimation [Williams, 1990], and for control and identification [Ku and Lee, 1995].