Chapter 1

Introduction and Role of Artificial Neural Networks

Artificial neural networks are, as their name indicates, computational networks which attempt to simulate, in a gross manner, the networks of nerve cell (neurons) of the biological (human or animal) central nervous system. This simulation is a gross cell-by-cell (neuron-by-neuron, element-by-element) simulation. It borrows from the neurophysiological knowledge of biological neurons and of networks of such biological neurons. It thus differs from conventional (digital or analog) computing machines that serve to replace, enhance or speed-up human brain computation without regard to organization of the computing elements and of their networking. Still, we emphasize that the simulation afforded by neural networks is very gross.

Why then should we view artificial neural networks (denoted below as neural networks or ANNs) as more than an exercise in simulation? We must ask this question especially since, computationally (at least), a conventional digital computer can do everything that an artificial neural network can do.

The answer lies in two aspects of major importance. The neural network, by its simulating a biological neural network, is in fact a novel computer architecture *and* a novel algorithmization architecture relative to conventional computers. *It allows using very simple computational operations* (additions, multiplication and fundamental logic elements) to solve complex, mathematically ill-defined problems, nonlinear problems or stochastic problems. A conventional algorithm will employ complex sets of equations, and will apply to only a given problem and exactly to it. *The ANN will be* (a) *computationally* and algorithmically *very simple* and (b) it will have a *self-organizing feature* to allow it to hold for a wide range of problems.

For example, if a house fly avoids an obstacle or if a mouse avoids a cat, it certainly solves no differential equations on trajectories, nor does it employ complex pattern recognition algorithms. Its brain is very simple, yet it employs a few basic neuronal cells that fundamentally obey the structure of such cells in advanced animals and in man. The artificial neural network's solution will also aim at such (most likely not the same) simplicity. Albert Einstein stated that a solution or a model must be as simple as possible to fit the problem at hand. Biological systems, in order to be as efficient and as versatile as they certainly are despite their inherent slowness (their basic computational step takes about a millisecond versus less than a nanosecond in today's electronic computers), can only do so by converging to the simplest algorithmic architecture that is possible. Whereas high level mathematics and logic can yield a broad general frame for solutions and can be reduced to specific but complicated algorithmization, the neural network's design aims at utmost simplicity and utmost self-organization. A very simple base algorithmic structure lies behind a neural network, but it is one which is highly adaptable to a broad range of problems. We note that at the present state of neural networks their range of adaptability is limited. However, their design is guided to achieve this simplicity and self-organization by its gross simulation of the biological network that is (must be) guided by the same principles.

Another aspect of ANNs that is different and advantageous to conventional computers, at least potentially, is in *its high parallelity* (element-wise parallelity). A conventional digital computer is a *sequential machine*. If one transistor (out of many millions) fails, then the whole machine comes to a halt. In the adult human central nervous system, neurons in the thousands die out each year, whereas brain function is totally unaffected, except when cells at very few key locations should die and this in very large numbers (e.g., major strokes). This insensitivity to damage of few cells is due to the high parallelity of biological neural networks, in contrast to the said sequential design of conventional digital computers (or analog computers, in case of damage to a single operational amplifier or disconnections of a resistor or wire). The same redundancy feature applies to ANNs. However, since presently most ANNs are still simulated on conventional digital computers, this aspect of insensitivity to component failure does not hold. Still, there is an increased availability of ANN hardware in terms of integrated circuits consisting of hundreds and even thousands of ANN neurons on a single chip does hold. [cf. Jabri *et al.*, 1996, Hammerstom, 1990, Haykin, 1994]. In that case, the latter feature of ANNs.

In summary, the excitement in ANNs should not be limited to its greater resemblance to the human brain. Even its degree of self-organizing capability can be built into conventional digital computers using complicated artificial intelligence algorithms. The main contribution of ANNs is that, in its gross imitation of the biological neural network, it allows for very low level programming to allow solving complex problems, especially those that are non-analytical and/or nonlinear and/or nonstationary and/or stochastic, *and* to do so in a self-organizing manner that applies to a wide range of problems with no re-programming or other interference in the program itself. The insensitivity to partial hardware failure is another great attraction, but only when dedicated ANN hardware is used.

It is becoming widely accepted that the advent of ANN will open *new understanding into how* to *simplify* programming and algorithm design for a given end and for a wide range of ends. It should bring attention to the simplest algorithm *without*, of course, *dethroning advanced mathematics* and logic, whose role will always be supreme in mathematical understanding and which will always provide a systematic basis for eventual reduction to specifics.

What is always amazing to many students and to myself is that after six weeks of class, first year engineering graduate students of widely varying backgrounds with no prior background in neural networks or in signal processing or pattern recognition, were able to solve, individually and unassisted, problems of speech recognition, of pattern recognition and character recognition, which could adapt in seconds or in minutes to changes (with a range) in pronunciation or in pattern. They would, by the end of the one-semester course, all be able to demonstrate these programs running and adapting to such changes, using PC simulations of their respective ANNs. My experience is that the study time and the background to achieve the same results by conventional methods by far exceeds that achieved with ANNs.

This, to me, demonstrates the degree of simplicity and generality afforded by ANN; and therefore the potential of ANNs.

Obviously, if one is to solve a set of differential equations, one would not use an ANN, just as one will not ask the mouse or the cat to solve it. But problems of recognition, filtering and control would be problems suited for ANNs. As always, no tool or discipline can be expected to do it all. And then, ANNs are certainly at their infancy. They started in the 1950s; and widespread interest in them dates from the early 1980s. So, all in all, ANNs deserve our serious attention. The days when they were brushed off as a gimmick or as a mere mental exercise are certainly over. Hybrid ANN/serial computer designs should also be considered to utilize the advantages of both designs where appropriate.