APPENDICES

APPENDIX A

The Programmable Schema

Digital computers are capable of being programmed, which means that they can have embedded in them a number of different alternative sets of instructions to perform various operations. Those operations can include among others, for example: data copying and transfer, mathematics, decisions among multiple alternative actions, and searching through data.

The structure of the digital computer by which all of its operations are performed consists of simple fundamental components: the “flip flop” also known as a bi-stable multivibrator and Boolean Algebra “gates” having the AND, OR, and NOT functions the action of those gates having been described in the preceding Section 2.

The gates perform their operations on the data. The data is represented in the two stable alternative states that the “flip flops” can have, the overall system being “binary”. that is all characters of data are represented by a number in a number system using the base *2* and the digits *0 and 1*. Of course, the number of flip flop’s and gates and the complexity of their interconnections in a digital computer is quite large.

THE NEURAL NETWORK AS A DIGITAL COMPUTER

The same type components as those that make the programmable digital computer possible are found in the brain's neural networks. The neuron effectively has two alternative stable states, able to fire [threshold exceeded] versus unable to fire [threshold too high]. The logic for the neuron is not Boolean Algebraic as in the digital computer; rather it is majority logic, but it was shown in Section 4 that majority logic with its use of constants can reproduce any Boolean Algebraic logic formulation.

For those reasons a brain's neural networks are fully as capable of being programmed, and of supporting operations per those programs as any digital computer except for one important difference.

- Humans must program digital computers, that is humans must design the program making it fit to the computer's physical structure and then embed the program into the computer.
- But, neural net computers must program themselves "on the job".

They must learn from experience by trial and error. It has to be that way because there is no alternative. On the other hand, that experience method produces the ultimately best adapted solution to the problem that it addresses.

One's *Master Schema* [MS] is evolved over one's lifetime. "Running things" since early childhood, since shortly before birth, your MS has learned what produces good results for you and what fails. One's various *sub-schema* have evolved by learning similar lessons. And life experiences have resulted in large enough modifications of existing *sub-schema* to their becoming newly designed additional new *sub-schema* a multiplying of the number of available *schema.*.

The *Master Schema* started with some of its neurons having initial thresholds that they were created with by the process of the development of the embryo under the "guidance" of its genes. Thus the beginning of the MS was the same as the beginning of other instincts. Thereafter experience, that is simply being alive in the world, "taught" the neural networks their programs.

"BOOTING UP"

A digital computer starts its day with power being turned on. The computer's programming then reaches down, grasps and pulls up on its own "boot straps", and lifts itself up. It "Boots Up", usually taking a small number of minutes to do so.

We animals having brains do more or less the same thing. We "Boot Up". We turn ourselves on, lifting ourselves up "by our own bootstraps". It is just that we don't do it daily but rather once in our life and for us it is a much longer drawn out process, part of our "growing up".